

ACCELERATING STOCHASTIC COLLOCATION METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS WITH RANDOM INPUT DATA *

D. GALINDO[†], P. JANTSCH[‡], C. G. WEBSTER[§], AND G. ZHANG[¶]

Abstract. This work proposes and analyzes a generalized acceleration technique for decreasing the computational complexity of using stochastic collocation (SC) methods to solve partial differential equations (PDEs) with random input data. The SC approaches considered in this effort consist of sequentially constructed multi-dimensional Lagrange interpolant in the random parametric domain, formulated by collocating on a set of points so that the resulting approximation is defined in a hierarchical sequence of polynomial spaces of increasing fidelity. Our acceleration approach exploits the construction of the SC interpolant to accelerate the underlying ensemble of deterministic solutions. Specifically, we predict the solution of the parametrized PDE at each collocation point on the current level of the SC approximation by evaluating each sample with a previously assembled lower fidelity interpolant, and then use such predictions to provide deterministic (linear or nonlinear) iterative solvers with improved initial approximations. As a concrete example, we develop our approach in the context of SC approaches that employ sparse tensor products of globally defined Lagrange polynomials on nested one-dimensional Clenshaw-Curtis abscissas. This work also provides a rigorous computational complexity analysis of the resulting fully discrete sparse grid SC approximation, with and without acceleration, which demonstrates the effectiveness of our proposed methodology in reducing the total number of iterations of a conjugate gradient solution of the finite element systems at each collocation point. Numerical examples include both linear and nonlinear parametrized PDEs, which are used to illustrate the theoretical results and the improved efficiency of this technique compared with several others.

Key words. stochastic and parametric PDEs, stochastic collocation, high-dimensional approximation, uncertainty quantification, sparse grids, multivariate polynomial approximation, iterative solvers, conjugate gradient method

AMS subject classifications. 65N30, 65N35, 65N12, 65N15, 65C20

1. Introduction. Modern approaches for predicting the behavior of physical and engineering problems, and assessing risk and informing decision making in manufacturing, economic forecasting, public policy, and human welfare, rely on mathematical modeling followed by computer simulation. Such predictions are obtained by constructing models whose solutions describe the phenomenon of interest, and then using computational methods to approximate the outputs of the models. Thus, the solution of a mathematical model can be viewed as a mapping from available input information onto a desired output of interest; predictions obtained through computational simulations are merely approximations of the images of the inputs, that is, of the output of interest. There are several causes for possible discrepancies between observations and approximate solutions obtained via computer simulations. The mathematical model may not, and usually does not, provide a totally faithful description of the phenomenon being modeled. Additionally, when an application is considered, the mathematical models need to be provided with input data, such as coefficients, forcing terms, initial and boundary conditions, geometry, etc. This input data may be affected by a large amount

*This material is based upon work supported in part by the U.S. Air Force of Scientific Research under grant number 1854-V521-12; by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract numbers ERKJ259, and ERKJE45; and by the Laboratory Directed Research and Development program at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC, for the U.S. Department of Energy under Contract DE-AC05-00OR22725.

[†]Joint Institute for Computational Sciences, University of Tennessee, 1 Bethel Valley Road, Oak Ridge, TN 37831 (dgalindi@utk.edu)

[‡]Department of Mathematics, University of Tennessee, Knoxville, TN 37996 (pjantsch@vols.utk.edu)

[§]Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN 37831 (webstercg@ornl.gov).

[¶]Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN 37831 (zhangg@ornl.gov).

of uncertainty due to intrinsic variability or the difficulty in accurately characterizing the physical system.

Such uncertainties can be included in the mathematical model by adopting a probabilistic setting, provided enough information is available for a complete statistical characterization of the physical system. In this effort we assume our mathematical model is described by a partial differential equation (PDE) and the random input data are modeled as finite dimensional random fields, parameterized by a vector $\mathbf{y} = (y_1, \dots, y_N)$ of dimension N , consisting of uncorrelated real-valued random variables. Therefore, the goal of the mathematical and computational analysis becomes the approximation of the solution map $\mathbf{y} \mapsto u(\mathbf{y})$, or statistical moments (mean, variance, covariance, etc.) of the solution or some quantity of interest (QoI) of the system, given the probability distribution of the input random data. A major challenge associated with developing approximation techniques for such problems involves alleviating the *curse of dimensionality*, by which the computational complexity of any naïve polynomial approach will grow exponentially with the dimension N of the parametric domain.

Monte Carlo (MC) methods (see, e.g., [17]) are the most popular approaches for approximating high-dimensional integrals, based on independent realizations $u(\mathbf{y}_k)$, $k = 1, \dots, M$, of the parameterized PDE; approximations of the expectation or other QoIs are obtained by averaging over the corresponding realizations of that quantity. The resulting numerical error is proportional to $M^{-1/2}$, thus achieving convergence rates independent of dimension N , but requiring a very large number of samples to achieve reasonably small errors. Other ensemble-based methods, including quasi-MC (QMC) and important sampling (see [24, 29, 39] and the references therein), have been devised to produce increase convergence rates, e.g., proportional to $M^{-1} \log(M)^{r(N)}$, however, the function $r(N) > 0$ increases with dimension N . Moreover, since both MC and QMC are quadrature techniques for QoIs, neither have the ability to simultaneously approximate the solution map $\mathbf{y} \mapsto u(\mathbf{y})$, required by a large class of applications.

In the last decade, two global polynomial approaches have been proposed that often feature much faster convergence rates: *intrusive* stochastic Galerkin (SG) methods, constructed from pre-defined orthogonal polynomials [19, 44], or best M -term and quasi-optimal approaches [6, 9, 12, 14], and *non-intrusive* stochastic collocation (SC) methods, constructed from (sparse) Lagrange interpolating polynomials [1, 30, 31], or discrete L^2 projections [27, 28]. These methods exploit the underlying regularity of the PDE solution map $u(\mathbf{y})$ with respect to the parameters \mathbf{y} , evident in a wide class of high-dimensional applications, to construct an approximate solution, and differ only in the choice of basis.

For both SG and SC approaches, the overall computational cost grows rapidly with increasing dimension. A recent development for alleviating such complexity and accelerating the convergence of parameterized PDE solutions is to utilize multilevel methods (see e.g., multilevel Monte Carlo (MLMC) methods [4, 5, 11, 20, 40] and the multilevel stochastic collocation (MLSC) approach [41]). The main ingredient to multilevel methods is the exploitation of a hierarchical sequence of spatial approximations to the underlying PDE, which are then combined with discretizations in parameter space in such a way as to minimize the overall computational cost. The approximation of the solution u on the finest mesh is represented by the approximation on the coarsest mesh plus a sequence of “correction” terms. The resulting decrease in complexity with the use of multilevel methods results from the fact that the dominant behavior of the solution u can be captured with cheap simulations on coarse meshes, so that the number of expensive simulations computed on fine meshes can be considerably reduced.

Nonetheless, the dominant cost in applying any uncertainty quantification (UQ) approach lies in the solution of the underlying parametrized linear/nonlinear PDEs, for a given value of the random inputs. Such solutions are often computed using iterative solvers, e.g., conjugate gradient

(CG) methods for symmetric positive-definite linear systems, generalized minimal residual method (GMRES) for non-symmetric linear systems [37], and fixed-point iteration methods [36] for nonlinear PDEs. However, many high-fidelity, multi-physics models can exhaust the resources of the largest machines with a single instantiation and, as such, are not practical for even the most advanced UQ techniques. As such, several methods for improving the performance of iterative solvers have been proposed; especially preconditioner and subspace methods for iterative Krylov solvers. A strategy that utilizes shared search directions for solving a collection of linear systems based on the CG method is proposed in [8]. In [33] a technique called *Krylov recycling* was introduced to solve sets of linear systems sequentially, based on ideas adapted from restarted and truncated GMRES (see [38] and the references therein). This approach was later applied to the linear systems that arise from SG approximations that use the so-called *doubly orthogonal* bases to solve stochastic parameterized PDEs [25]. In addition, several preconditioners have been developed that improve the performance of solving the large linear systems resulting from SG approximations that employ standard orthogonal polynomials [16, 18, 21, 35].

On the other hand, when a general linear solver is employed to solve the underlying SG or SC approximation, it is straightforward to see that improved initial approximations can significantly reduce the number of iterations required to reach a prescribed accuracy. A sequential orthogonal expansion is utilized in [18, 34] such that a low resolution solution provides an initial guess for the solution of the system with an enriched basis. However, at each step, all the expansion coefficients must be explicitly recomputed, resulting in increased costs. Similarly, in [21] an extension of a mean-based preconditioner is applied to each linear system coming from a sequential SC approach, wherein the solution of the j -th system is given as the initial vector for the $(j + 1)$ -th system. This approach, as well as the Krylov recycling method, impose an ordering of the linear systems that appear in the SC approximation. Consequently, new approaches are needed to amortize the cost of expensive simulations by reusing both deterministic and stochastic information across multiple ensembles of solutions.

In this work, we propose to improve the computational efficiency of non-intrusive approximations, by focusing on SC approaches that sequentially construct a multi-dimensional Lagrange interpolant in a hierarchical sequence of polynomial spaces of increasing fidelity. As opposed to multilevel methods that reduce the overall computational burden by taking advantage of a hierarchical spatial approximation, our approach exploits the structure of the SC interpolant to accelerate the underlying ensemble of deterministic solutions. Specifically, we predict the solution of the parametrized PDE at each collocation point on the current level of the SC approximation by evaluating each sample with a previously assembled lower fidelity interpolant, and then use such predictions to provide deterministic (linear or nonlinear) iterative solvers with improved initial approximations. As a particular application, we pose this acceleration technique in the context of hierarchical SC methods that employ sparse tensor products of globally defined Lagrange polynomials [30, 31], on nested one-dimensional Clenshaw-Curtis abscissas. However, the same idea can be extended to other non-intrusive collocation approaches including orthogonal polynomials [44], as well as piecewise local and wavelet polynomials expansions [7, 22].

The sparse grid SC approximation considered in this work produces a sequence of interpolants, where a new set of collocation points is added on each level in order to increase the accuracy of the interpolant. For each newly added collocation point on the current level, we predict the solution of the underlying deterministic PDE using the most up to date sparse grid interpolant available; the previous level's interpolant. We then use the prediction as the starting point of the iterative solver. The uniform convergence of the sparse grid interpolant to the true solution results in an increasingly accurate initial guess as the level increases, so that the overall complexity of the SC method can be dramatically reduced. We apply our novel approach in the context of solving both linear and

nonlinear stochastic PDEs, wherein, we assume that the parameterized systems are solved by some existing linear or nonlinear iterative method. Furthermore, in the linear case, this technique can also be used to efficiently generate improved preconditioners for linear systems associated to the collocation points on higher levels, which further accelerates the convergence rate of the underlying solver.

The outline of this paper is as follows: We begin by describing the class of parameterized linear and nonlinear stochastic PDEs under consideration in §2. In §3 we describe our acceleration technique in the context of general stochastic collocation methods, defined on a hierarchical sequence of polynomial spaces, for approximating both linear and nonlinear stochastic elliptic PDEs using nonlinear iterative solvers. In §4 we briefly recall the sparse grid SC method, where the sparse grid interpolant is constructed with the use of nested one-dimensional Clenshaw-Curtis abscissas. The theoretical convergence rates, with respect to the level of the interpolant and the degrees of freedom are shown in §4.1. In §4.2 we provide a rigorous computational complexity analysis of the resulting fully discrete sparse grid SC approximation, with and without acceleration, used to demonstrate the effectiveness of our proposed methodology in reducing the total number of iterations of a conjugate gradient solution of the finite element systems at each collocation point. Finally, in §5 we provide several numerical examples, including both moderately large-dimensional linear and nonlinear parametrized PDEs, which are used to illustrate the theoretical results and the improved efficiency of this technique compared with several others.

2. Problem setting. Let $D \subset \mathbb{R}^d$, $d = 1, 2, 3$, be a bounded domain and let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space with sample space Ω , σ -algebra $\mathcal{F} = 2^\Omega$, and probability measure $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$. Define \mathcal{L} as a differential operator that depends on a coefficient $a(x, \omega)$ with $x \in D$ and $\omega \in \Omega$. Analogously, the forcing term $f = f(x, \omega)$ can be assumed to be a random field as well. In general, a and f belong to different probability spaces but, for economy of notation, we simply denote the stochastic dependences in the same probability space. Consider the stochastic boundary value problem. Find a random function $u : \overline{D} \times \Omega \rightarrow \mathbb{R}$ such that, \mathbb{P} -a.e. in Ω , the following equations hold:

$$\begin{cases} \mathcal{L}(a)(u) = f & \text{in } D, \\ u = g & \text{on } \partial D, \end{cases} \quad (2.1)$$

where g is a suitable boundary condition. We denote by $W(D)$ a Banach space and assume the underlying random input data are chosen so that the corresponding stochastic system (2.1) is well-posed and has a unique solution $u(x, \omega) \in L^q_{\mathbb{P}}(\Omega; W(D))$, the function space given by

$$L^q_{\mathbb{P}}(\Omega; W(D)) := \left\{ u : \overline{D} \times \Omega \rightarrow \mathbb{R} \mid u \text{ is strongly measurable and } \int_{\Omega} \|u\|_{W(D)}^q d\mathbb{P}(\omega) < +\infty \right\}.$$

In this setting, the approximation space consists of Banach-space valued functions that have finite q -th order moments. Two example problems posed in this setting are given as follows.

EXAMPLE 2.1. (Linear elliptic problem). *Find a random field $u : \overline{D} \times \Omega \rightarrow \mathbb{R}$ such that \mathbb{P} -a.e.*

$$\begin{cases} -\nabla \cdot (a(x, \omega) \nabla u(x, \omega)) & = f(x, \omega) & \text{in } D \times \Omega, \\ u(x, \omega) & = 0 & \text{on } \partial D \times \Omega, \end{cases} \quad (2.2)$$

where ∇ denotes the gradient operator with respect to the spatial variable $x \in D$. The well-posedness of (2.2) is guaranteed in $L^2_{\mathbb{P}}(\Omega; H^1_0(D))$ with $a(x, \omega)$ uniformly elliptic, i.e.,

$$\mathbb{P}(\omega \in \Omega : a_{\min} \leq a(x, \omega) \leq a_{\max} \ \forall x \in \overline{D}) = 1 \text{ with } a_{\min}, a_{\max} \in (0, \infty), \quad (2.3)$$

and $f(x, \omega)$ square integrable, i.e.,

$$\int_D \mathbb{E}[f^2] dx := \int_D \int_{\Omega} f^2(x, \omega) d\mathbb{P}(\omega) dx < +\infty.$$

EXAMPLE 2.2. (Nonlinear elliptic problem). For $k \in \mathbb{N}$, find a random field $u : \overline{D} \times \Omega \rightarrow \mathbb{R}$ such that \mathbb{P} -a.e.

$$\begin{cases} -\nabla \cdot (a(x, \omega) \nabla u(x, \omega)) + u(x, \omega) |u(x, \omega)|^k &= f(x, \omega) & \text{in } D, \\ u(x, \omega) &= 0 & \text{on } \partial D. \end{cases} \quad (2.4)$$

The well-posedness of (2.4) is guaranteed in $L^2_{\mathbb{P}}(\Omega; W(D))$ with a, f as in Example 2.1 and $W(D) = H^1_0(D) \cap L^{k+2}(D)$ [31].

In many applications, the source of randomness can be approximated with only a finite number of uncorrelated, or even independent, random variables. For instance, the random input data a and f in (2.1) may have a piecewise representation, or in other applications may have spatial variation that can be modeled as a correlated random field, making them amenable to approximation by a Karhunen-Loève (KL) expansion [26]. In practice, one has to truncate such expansions according to the desired accuracy of the simulation. As such, we make the following assumption regarding the random input data a and f (cf [23, 31]).

ASSUMPTION 2.1. (Independence and finite dimensional noise). The random fields $a(x, \omega)$ and $f(x, \omega)$ have the form:

$$a(x, \omega) = a(x, \mathbf{y}(\omega)) \text{ and } f(x, \omega) = f(x, \mathbf{y}(\omega)) \text{ on } D \times \Omega,$$

where $\mathbf{y}(\omega) = [y_1(\omega), \dots, y_N(\omega)] : \Omega \rightarrow \mathbb{R}^N$ is a vector of independent and uncorrelated real-valued random variables.

We note that Assumption 2.1 and the Doob-Dynkin lemma [32] guarantee that $a(x, \mathbf{y}(\omega))$ and $f(x, \mathbf{y}(\omega))$ are Borel-measurable functions of the random vector $\mathbf{y} : \Omega \rightarrow \mathbb{R}^N$. In our setting, we denote by $\Gamma_n = y_n(\Omega) \subset \mathbb{R}$ the image of the random variable y_n , and set $\Gamma = \prod_{n=1}^N \Gamma_n$, where $N \in \mathbb{N}_+$. If the distribution measure of $\mathbf{y}(\omega)$ is absolutely continuous with respect to Lebesgue measure, then there exists a joint probability density function of $\mathbf{y}(\omega)$ denoted by

$$\varrho(\mathbf{y}) : \Gamma \rightarrow \mathbb{R}_+, \text{ with } \varrho(\mathbf{y}) = \prod_{n=1}^N \varrho_n(y_n) \in L^\infty(\Gamma).$$

Therefore, based on Assumption 2.1, the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is mapped to $(\Gamma, \mathcal{B}(\Gamma), \varrho(\mathbf{y}) d\mathbf{y})$, where $\mathcal{B}(\Gamma)$ is the Borel σ -algebra on Γ and $\varrho(\mathbf{y}) d\mathbf{y}$ is a probability measure on $\mathcal{B}(\Gamma)$. By assuming the solution u of (2.1) is σ -measurable with respect to a and f , the Doob-Dynkin lemma guarantees that $u(x, \omega)$ can also be characterized by the same random vector \mathbf{y} , i.e.,

$$u(x, \omega) = u(x, y_1(\omega), \dots, y_N(\omega)) \in L^q_{\varrho}(\Gamma; W(D)),$$

where $L^q_\varrho(\Gamma; W(D))$ is defined by

$$L^q_\varrho(\Gamma; W(D)) = \left\{ u : \overline{D} \times \Gamma \rightarrow \mathbb{R} \mid u \text{ strongly measurable and } \int_\Gamma \|u\|_{W(D)}^q \varrho(\mathbf{y}) d\mathbf{y} < \infty \right\}.$$

Note that the above integral will be replaced by the essential supremum when $q = \infty$:

$$L^\infty(\Gamma; W(D)) = \left\{ u : \overline{D} \times \Gamma \rightarrow \mathbb{R} \mid u \text{ strongly measurable and } \text{ess sup}_{\mathbf{y}} \|u(\mathbf{y})\|_{W(D)} < \infty \right\}.$$

2.1. Weak formulation. In what follows, we treat the solution to (2.1) as a parameterized function $u(x, \mathbf{y})$ of the N -dimensional random variables $\mathbf{y} \in \Gamma \subset \mathbb{R}^N$. This leads to a Galerkin weak formulation [23] of the PDE in (2.1), with respect to both physical and parameter space, i.e., seek $u \in L^q_\varrho(\Gamma; W(D))$ such that

$$\int_\Gamma \int_D \left(\sum_{\nu \in \Lambda_1 \cup \Lambda_2} S_\nu(u; \mathbf{y}) T_\nu(v) \right) \varrho dx d\mathbf{y} = \int_\Gamma \int_D f v \varrho dx d\mathbf{y}, \quad \forall v \in L^q_\varrho(\Gamma; W(D)),$$

where $T_\nu, \nu \in \Lambda_1 \cup \Lambda_2$ are linear operators independent of \mathbf{y} , while the operators S_ν are linear for $\nu \in \Lambda_1$, and nonlinear for $\nu \in \Lambda_2$. Moreover, since the solution u can be viewed as a mapping $u : \Gamma \rightarrow W(D)$, for convenience we may omit the dependence on $x \in D$ and write $u(\mathbf{y})$ to emphasize the dependence of u on \mathbf{y} . As such, we may also write the problem (2.1) in the alternative weak form

$$\int_D \left(\sum_{\nu \in \Lambda_1 \cup \Lambda_2} S_\nu(u(\mathbf{y}); \mathbf{y}) T_\nu(v) \right) dx = \int_D f(\mathbf{y}) v dx, \quad \forall v \in W(D), \quad \varrho\text{-a.e. in } \Gamma. \quad (2.5)$$

Therefore, the stochastic boundary-value problem (2.1) has been converted into a deterministic parametric problem (2.5). The acceleration technique proposed in §3 and the sparse-grid SC method discussed in §4 will be based on the solution of the weak form (2.5) above.

3. Accelerating stochastic collocation methods. Our acceleration scheme will be proposed in the context of both linear and nonlinear elliptic PDEs. A general SC approach requires the semi-discrete solution $u_h(\cdot, \mathbf{y}) \in W_h(D) \subset W(D)$ at a set of collocation points $\{\mathbf{y}_{L,j}\}_{j=1}^{M_L} \subset \Gamma$, given by

$$u_h(x, \mathbf{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x), \quad j = 1, \dots, M_L. \quad (3.1)$$

Here $\{\varphi_i\}_{i=1}^{M_h}$ is a predefined finite element basis of $W_h(D)$, and for $j = 1, \dots, M_L$, the coefficient vector $\mathbf{c}_{L,j} := (c_{L,j,1}, \dots, c_{L,j,M_h})^\top$ is the solution of the following system of equations:

$$\begin{aligned} & \sum_{i=1}^{M_h} c_{L,j,i} \int_D \sum_{\nu \in \Lambda_1} S_\nu(\varphi_i; \mathbf{y}_{L,j}) T_\nu(\varphi_{i'}) dx \\ &= \int_D f(\mathbf{y}_{L,j}) \varphi_{i'} - \sum_{\nu \in \Lambda_2} S_\nu \left(\sum_{i=1}^{M_h} c_{L,j,i} \varphi_i; \mathbf{y}_{L,j} \right) T_\nu(\varphi_{i'}) dx, \quad i' = 1, \dots, M_h, \end{aligned} \quad (3.2)$$

with S_ν and T_ν defined as above. Note that (3.2) is equivalent to (2.5) with the nonlinear operators subtracted on the right hand side. When $\Lambda_2 = \emptyset$, the PDE is linear, and a standard FEM discretization leads to a linear system of equations.

For $L \in \mathbb{N}_+$, we denote by \mathcal{I}_L an interpolation operator that utilizes M_L collocation points, defined by $\mathcal{H}_L = \{\mathbf{y}_{L,j}\}_{j=1}^{M_L}$. More generally, assume that we have a family of interpolation operators $\{\mathcal{I}_L\}_{L \in \mathbb{N}_+}$, which for each $L \in \mathbb{N}_+$ approximates the solution $u_h(x, \cdot)$ in polynomial spaces

$$\mathcal{P}_1(\Gamma) \subset \dots \subset \mathcal{P}_L(\Gamma) \subset \mathcal{P}_{L+1}(\Gamma) \subset \dots \subset L_\theta^2(\Gamma),$$

of increasing fidelity, defined on sets of sample points $\mathcal{H}_L \subset \Gamma$. Assume further that the fully discrete solution $u_{h,L} \in W_h(D) \otimes \mathcal{P}_L(\Gamma)$ has Lagrange interpolating form

$$u_{h,L}(x, \mathbf{y}) := \mathcal{I}_L[u_h](x, \mathbf{y}) = \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x) \right) \Psi_{L,j}(\mathbf{y}), \quad (3.3)$$

where $\{\Psi_{L,j}\}_{j=1}^{M_L}$ is a basis for $\mathcal{P}_L(\Gamma)$. The approximation (3.3) can be constructed by solving for $u_h(x, \mathbf{y}_{L,j})$ *independently* at each sample point $\mathbf{y}_{L,j} \in \mathcal{H}_L$. In §4, we construct a specific example of an interpolation scheme satisfying (3.3), namely global sparse grid collocation.

For each $L \in \mathbb{N}$, the bulk of the computational cost in using (3.3) goes into solving the M_L systems of equations (3.2) corresponding to each collocation point $\mathbf{y}_{L,j}$, $j = 1, \dots, M_L$. Since the systems are independent and deterministic, they can be solved separately using existing FEM solvers, providing a straightforward path to parallelization compared to intrusive methods such as stochastic Galerkin methods. In this work, we consider iterative solvers for the system in (3.2), and propose an acceleration scheme to reduce the total number of iterations necessary to the collection of systems over the set of sample parameters.

Denoting by \tilde{u}_h the output of the selected iterative solver for the system (3.2), for $\mathbf{y}_{L,j} \in \mathcal{H}_L$ the semi-discrete solution $u_h(x, \mathbf{y}_{L,j})$ is approximated by

$$u_h(x, \mathbf{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x) \approx \tilde{u}_h(x, \mathbf{y}_{L,j}) = \sum_{i=1}^{M_h} \tilde{c}_{L,j,i} \varphi_i(x),$$

where we define $\tilde{\mathbf{c}}_{L,j} = (\tilde{c}_{L,j,1}, \dots, \tilde{c}_{L,j,M_h})^\top$, and therefore the final SC approximation is given by a perturbation of (3.3), i.e.,

$$\tilde{u}_{h,L}(x, \mathbf{y}) := \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} \tilde{c}_{L,j,i} \varphi_i(x) \right) \Psi_{L,j}(\mathbf{y}). \quad (3.4)$$

We observe that the performance of the underlying iterative solver can be improved by proposing a good initial guess, denoted $\mathbf{c}_{L,j}^{(0)}$, or constructing an effective preconditioner to reduce the condition number of the system. Here, we propose our approach for improving initial deterministic approximations, remarking that the same idea can be also utilized to construct preconditioners. To start the iterative solver for the system in (3.2), it is common to use a zero initial guess, i.e., $\mathbf{c}_{L,j}^{(0)} = (0, \dots, 0)^\top$. However, we can predict the solution at level L using lower level approximations to construct improved initial solutions $\mathbf{c}_{L,j}^{(0)}$. Assume that we first obtain $\tilde{u}_{h,L-1}(x, \mathbf{y})$ by collocating solutions to (3.2) over \mathcal{H}_{L-1} . Then at level L , for each new point $\mathbf{y}_{L,j} \in \mathcal{H}_L \setminus \mathcal{H}_{L-1}$, the initial guess $\mathbf{c}_{L,j}^{(0)}$ can

be given by interpolating the solutions from level $L - 1$, i.e.,

$$\mathbf{c}_{L,j}^{(0)} = \left(\tilde{u}_{h,L-1}(x_1, \mathbf{y}_{L,j}), \dots, \tilde{u}_{h,L-1}(x_{M_h}, \mathbf{y}_{L,j}) \right)^\top = \sum_{j'=1}^{M_{L-1}} \tilde{\mathbf{c}}_{L-1,j'} \Psi_{L-1,j'}(\mathbf{y}_{L,j}). \quad (3.5)$$

For a convergent interpolation scheme, we expect the necessary number of iterations to compute $\tilde{\mathbf{c}}_{L,j}$ to become smaller as the level L increases to an overall maximum level, denoted L_{\max} . As such, the construction of the desired solution $\tilde{u}_{h,L_{\max}}$ is accelerated through the intermediate solutions $\{\tilde{u}_{h,L}\}_{L=1}^{L_{\max}-1}$. Note that this approach reduces computational cost by improving initial guesses, but does not depend on the specific solver used. Thus, our scheme may be combined with other techniques for accelerating convergence, such as faster nonlinear solvers or better preconditioners. When the underlying PDE is nonlinear with respect to u , iterative solvers are commonly used for the solution of (3.2). In Algorithm 1, we outline the acceleration procedure described above, using a general nonlinear iterative method for the solution of (3.2).

Algorithm 1: *The accelerated SC algorithm*

Goal: Compute $\tilde{u}_{h,L_{\max}}(x, \mathbf{y}) := \sum_{i=1}^{M_{L_{\max}}} \left(\sum_{j=1}^{M_h} \tilde{\mathbf{c}}_{L_{\max},j,i} \varphi_i(x) \right) \Psi_{L_{\max},j}(\mathbf{y})$

```

1: Define  $M_0 = 1$  and  $\tilde{\mathbf{c}}_{0,1} = (0, \dots, 0)^\top$ 
2: for  $L = 1, \dots, L_{\max}$  do
3:   for  $\mathbf{y}_{L,j} \in \mathcal{H}_L \setminus \left( \bigcup_{l=1}^{L-1} \mathcal{H}_l \right)$  do
4:     Compute the initial guess according to (3.5):
5:      $\mathbf{c}_{L,j}^{(0)} = \sum_{j'=1}^{M_{L-1}} \tilde{\mathbf{c}}_{L-1,j'} \Psi_{L-1,j'}(\mathbf{y}_{L,j})$ 
6:     Initialize:  $k = 1$ 
7:     repeat
8:       Compute residual  $\mathbf{r}_{L,j}^{(k)} = (r_{L,j,1}^{(k)}, \dots, r_{L,j,M_h}^{(k)})^\top$ :
9:       for  $i = 1, \dots, M_h$  do
10:         $r_{L,j,i}^{(k)} = \int_D f(\mathbf{y}_{L,j}) \varphi_i - \sum_{\nu \in \Lambda_1 \cup \Lambda_2} S_\nu \left( \sum_{i'=1}^{M_h} \mathbf{c}_{L,j,i'}^{(k)} \varphi_{i'}(x), \mathbf{y}_{L,j} \right) T_\nu(\varphi_i) dx$ 
11:       end for
12:       Update the solution:  $\mathbf{c}_{L,j}^{(k+1)} = \mathbf{c}_{L,j}^{(k)} + \mathcal{S}(\mathbf{r}_{L,j}^{(1)}, \dots, \mathbf{r}_{L,j}^{(k)})$ 
13:        $k = k + 1$ 
14:     until  $\|\mathbf{c}_{L,j}^{(k)} - \mathbf{c}_{L,j}^{(k-1)}\| < \tau$ 
15:      $\tilde{\mathbf{c}}_{L,j} = \mathbf{c}_{L,j}^{(k)}$ 
16:   end for
17: end for
```

The efficiency of the proposed algorithm will depend crucially on the number of times the iterative solver is utilized, i.e., how many sample points are in the set $\Delta\mathcal{H}_L = \mathcal{H}_L \setminus \left(\bigcup_{l=1}^{L-1} \mathcal{H}_l \right)$ for each level L . In fact, if the sample points are not nested, it could be the case that $\Delta\mathcal{H}_L = \mathcal{H}_L$, and the algorithm may be very inefficient. Hence, in the following sections we will assume:

ASSUMPTION 3.1. *Assume that the multidimensional point sets $\mathcal{H}_L, L = 1, \dots, L_{\max}$ are nested,*

i.e.,

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots \subset \mathcal{H}_{L_{\max}} \subset \Gamma.$$

Then $\Delta\mathcal{H}_L = \mathcal{H}_L \setminus \mathcal{H}_{L-1}$, and we can construct the intermediate solutions $\{\tilde{u}_{h,L}\}_{L=1}^{L_{\max}-1}$ using a subset of the information needed to approximate $\tilde{u}_{h,L_{\max}}$.

In §4 we construct an interpolant using a point set which satisfies Assumption 3.1. Next, we give several examples of Algorithm 1, using iterative solvers for both nonlinear and linear elliptic PDEs.

EXAMPLE 3.1. Consider the weak form of the nonlinear elliptic PDE in Example 2.2, letting $S_1(v; \mathbf{y}) = a(x, \mathbf{y})\nabla v$, $T_1(v) = \nabla v$, $S_2(v, \mathbf{y}) = v(x, \mathbf{y})|v(x, \mathbf{y})|^s$, and $T_2(v) = v$ (note that $\Lambda_1 = \{1\}$, $\Lambda_2 = \{2\}$). When using the fixed point iterative method in Algorithm 1, for the update step we define

$$\mathcal{S}(\mathbf{r}_{L,j}^{(1)}, \dots, \mathbf{r}_{L,j}^{(k)}) = \mathbf{A}_{L,j}^{-1} \mathbf{r}_{L,j}^{(k)},$$

where the matrix $\mathbf{A}_{L,j} = \mathbf{A}(\mathbf{y}_{L,j})$, $j = 1, \dots, M_L$ is defined by

$$[\mathbf{A}_{L,j}]_{i,i'} = \int_D a(\mathbf{y}_{L,j}) \nabla \varphi_{i'} \nabla \varphi_i dx, \quad \text{for } i, i' = 1, \dots, M_h. \quad (3.6)$$

With $u_{h,L}^{(k)}(x, \mathbf{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i}^{(k)} \varphi_i(x)$, this update is equivalent to solving the following linear system

$$\int_D a(\mathbf{y}_{L,j}) \nabla u_{h,L}^{(k+1)} \nabla v dx = \int_D \left[f(\mathbf{y}_{L,j}) - u_{h,L}^{(k)}(\mathbf{y}_{L,j}) |u_{h,L}^{(k)}(\mathbf{y}_{L,j})|^s \right] v dx \quad \forall v \in W_h(D),$$

to update $u_h^{(k)}$ to $u_h^{(k+1)}$ at the $(k+1)$ -th iteration. Note that each iteration of the solver in Algorithm 1 requires the solution of this linear system, which is not accelerated by our algorithm.

EXAMPLE 3.2. As a special case of the example above, consider the weak form of the linear elliptic problem in Example 2.1 with $\Lambda_1 = \{1\}$, $\Lambda_2 = \emptyset$, $S_1(v; \mathbf{y}) = a\nabla v$ and $T_1(v) = \nabla v$ in (3.2). Due to the linearity, at each collocation point the solution $u_h(x, \mathbf{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x)$ can be approximated by solving the following linear system

$$\mathbf{A}_{L,j} \mathbf{c}_{L,j} = \mathbf{f}_{L,j}, \quad (3.7)$$

with $\mathbf{A}_{L,j} = \mathbf{A}(\mathbf{y}_{L,j})$, $j = 1, \dots, M_L$ as in (3.6), and $(\mathbf{f}_{L,j})_i = \int_D f(x, \mathbf{y}_{L,j}) \varphi_i(x) dx$ for $i = 1, \dots, M_h$. Under our assumptions on the coefficient a , the linear system (3.7) is symmetric positive definite, and we can use the CG method [37] to find its solution. For $k \in \mathbb{N}^+$, by recursively defining

$$\mathbf{p}_{L,j}^{(k)} = \mathbf{r}_{L,j}^{(k)} - \sum_{k' < k} \frac{\mathbf{p}_{L,j}^{(k')\top} \mathbf{A}_{L,j} \mathbf{r}_{L,j}^{(k)}}{\mathbf{p}_{L,j}^{(k')\top} \mathbf{A}_{L,j} \mathbf{p}_{L,j}^{(k')}} \mathbf{p}_{L,j}^{(k')},$$

we get the update function

$$\mathcal{S}(\mathbf{r}_{L,j}^{(1)}, \dots, \mathbf{r}_{L,j}^{(k)}) = \frac{\mathbf{p}_{L,j}^{(k)\top} \mathbf{r}_{L,j}^{(k)}}{\mathbf{p}_{L,j}^{(k)\top} \mathbf{A}_{L,j} \mathbf{p}_{L,j}^{(k)}} \mathbf{p}_{L,j}^{(k)}.$$

Recall the following well-known error estimate for CG:

$$\left\| \mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(k)} \right\|_{\mathbf{A}_{L,j}} \leq 2 \left(\frac{\sqrt{\kappa_{L,j}} - 1}{\sqrt{\kappa_{L,j}} + 1} \right)^k \left\| \mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(0)} \right\|_{\mathbf{A}_{L,j}}, \quad (3.8)$$

where $\kappa_{L,j} = \kappa(\mathbf{y}_{L,j})$ denotes the condition number of $\mathbf{A}_{L,j}$, $\mathbf{c}_{L,j}^{(0)}$ is the vector of initial guess and $\mathbf{c}_{L,j}^{(k)}$ is the output of the k -th iteration of the CG solver. As opposed to Example 3.1, for this example Algorithm 1 accelerates the solution of the linear system (3.7).

To evaluate the efficiency of the accelerated SC method, we define cost metrics for the construction of standard and accelerated SC approximations. In general, the computational cost in floating point operations (flops) is the total number iterations to solve (3.2) summed over each of the sample points—denoted by K_{zero} and K_{acc} for the standard and accelerated SC methods, respectively—multiplied by the cost of performing one iteration, denoted $\mathcal{C}_{\text{iter}}$. Let \mathcal{C}_{int} be the additional cost of interpolation incurred by using the accelerated initial vectors (3.5). Then, we define

$$\mathcal{C}_{\text{zero}} = \mathcal{C}_{\text{iter}} K_{\text{zero}}, \quad (3.9)$$

for the standard SC approach, and

$$\mathcal{C}_{\text{acc}} = \mathcal{C}_{\text{iter}} K_{\text{acc}} + \mathcal{C}_{\text{int}}, \quad (3.10)$$

for the accelerated SC approximation, respectively.

In Example 3.2 the discretization of the linear PDE leads to M_L sparse systems of equations of size $M_h \times M_h$. When solving these systems with a CG solver, K_{zero} and K_{acc} are the sum of solver iterations contributed from each sample system. In this case, the cost of one iteration is just the cost of one matrix vector product, i.e., $\mathcal{C}_{\text{iter}} = C_D M_h$, where C_D depends on the domain D and the type of finite element basis.

REMARK 3.1. (Relationship to multilevel methods). *Multilevel methods reduce the complexity of stochastic sampling methods by balancing errors and computational cost across a sequence of stochastic and spatial approximations. Let $u_{h_k} \in V_k$, $k = 0, \dots, K$, be a sequence of semi-discrete approximations built in nested spaces, i.e., $V_0 \subset \dots \subset V_K$. Multilevel methods are based on the following identity:*

$$u_{h_K} = \sum_{k=0}^K (u_{h_k} - u_{h_{k-1}}).$$

Letting $\mathcal{Q}_{L_{K-k}}$, $k = 0, \dots, K$, denote the chosen method of stochastic approximation, a general multilevel method might be written as

$$u_K^{(\text{ML})} = \sum_{k=0}^K \mathcal{Q}_{L_{K-k}} [u_{h_k} - u_{h_{k-1}}].$$

The main idea is that highly resolved, expensive stochastic approximations, e.g., \mathcal{Q}_{L_K} , in combination with coarse deterministic approximations, that is, u_{h_0} , and vice versa. In a similar way, collocation with nested grid points provides a natural multilevel hierarchy which we use in our method to accelerate each PDE solve (3.5). A combination of these methods could involve using our algorithm to accelerate the construction of the operators $\mathcal{Q}_{L_{K-k}}$, as well as reusing information from level to level, thus improving further the performance of SC methods.

REMARK 3.2. (Interpolation costs). Note that many adaptive interpolation schemes already require evaluation of the intermediate interpolation operators as in (3.5), e.g., to compute residual error estimators. Thus, these methods will incur the interpolation cost \mathcal{C}_{int} even in the case of zero initial vectors. Furthermore, for most nonlinear problems the deterministic solver is expensive, thus reducing the number of iterations is the most important element in reducing the cost. In each of these settings, we can define the cost metrics to simply be K_{zero} and K_{acc} .

REMARK 3.3. (Hierarchical preconditioner construction). When solving linear systems using iterative methods, convergence properties can be improved by considering the condition number of the system. As with initial vectors, an interpolation algorithm can be used to construct good, cheap preconditioners. We consider preconditioner algorithms where an explicit preconditioner matrix, or its inverse, is constructed. In this case, for some low collocation level L_{PC} , we construct a strong preconditioner, $P_{L_{\text{PC}},j} := P(\mathbf{y}_{L_{\text{PC}},j})$, for each individual iterative solver, $j = 1, \dots, M_{L_{\text{PC}}}$. Then, these lower level preconditioners are interpolated for the subsequent levels. More specifically, for $L > L_{\text{PC}}$, and $\mathbf{y}_{L,j} \in \mathcal{H}_L \setminus \mathcal{H}_{L_{\text{PC}}}$, we use the preconditioner

$$\tilde{P}_{L,j} := \tilde{P}(\mathbf{y}_{L,j}) = \sum_{j'=1}^{M_{L_{\text{PC}}}} P_{L_{\text{PC}},j'} \Psi_{L_{\text{PC}},j'}(\mathbf{y}_{L,j}). \quad (3.11)$$

Numerical illustrations of this approach are given in §5.

4. Applications to sparse grid stochastic collocation. In this section, we provide a specific example of an interpolation scheme satisfying the assumptions described in §3, i.e., a generalized sparse grid SC approach for a fixed level L . In what follows, we briefly review the construction of sparse grid interpolants, and rigorously analyze the approximation errors and the complexities of both the standard and accelerated SC approaches, in order to demonstrate the improved efficiency of the proposed acceleration technique when applied to iterative linear solvers.

The fully discrete SC approximation is built by polynomial interpolation of the semi-discrete solution $u_h(x, \mathbf{y})$ on an appropriate set of collocation points in Γ . In our setting, such an interpolation scheme is based on a sparse tensor products of one-dimensional Lagrange interpolating polynomials with global support. Specifically, in the one-dimensional case, $N = 1$, we introduce a sequence of Lagrange interpolation operators $\mathcal{U}^{m(l)} : C^0(\Gamma) \rightarrow \mathcal{P}_{m(l)-1}(\Gamma)$, with $\mathcal{P}_{m(l)-1}(\Gamma)$ the space of degree $m(l) - 1$ polynomials over Γ . Given a general function $v \in C^0(\Gamma)$, these operators are defined by

$$\mathcal{U}^{m(l)}[v](y) = \sum_{j=1}^{m(l)} v(y_j^l) \psi_j^l(y).$$

Here $l \in \mathbb{N}$ represents the resolution level of the operator, $m(l) \in \mathbb{N}_+$ denotes the number of interpolation points on level l , $\psi_1^1(y) = 1$ and for $l > 1$,

$$\psi_j^l(y) = \prod_{\substack{i=1 \\ i \neq j}}^{m(l)} \frac{y - y_i^l}{y_j^l - y_i^l} \quad \text{for } j = 1, \dots, m(l),$$

are the global Lagrange polynomials of degree $m(l)-1$ associated with the point set $\vartheta^l = \{y_1^l, \dots, y_{m(l)}^l\}$.

To satisfy Assumption 3.1, we need nestedness of the one-dimensional sets, i.e., $\vartheta^{l-1} \subset \vartheta^l$, which is determined by the choice of interpolation points and the definition of $m(l)$. In addition, we remark that similar constructions for $\mathcal{U}^{m(l)}$ can be built based on wavelets [22] or other locally supported

polynomial functions [23].

In the multi-dimensional case, i.e., $N > 1$, using the convention that $\mathcal{U}^{m(0)} = 0$, we introduce the difference operator

$$\Delta^{m(l_1)} \otimes \dots \otimes \Delta^{m(l_N)} = \bigotimes_{n=1}^N \left(\mathcal{U}^{m(l_n)} - \mathcal{U}^{m(l_n-1)} \right), \quad (4.1)$$

and define the multi-index $\mathbf{l} = (l_1, \dots, l_N) \in \mathbb{N}_+^N$. The desired approximation is defined by a linear combination of tensor-product operators (4.1) over a set of multi-indices, determined by the condition $g(\mathbf{l}) \leq L$, for $L \in \mathbb{N}_+$, and $g(\mathbf{l}) : \mathbb{N}_+^N \rightarrow \mathbb{N}_+$ a strictly increasing function. For $v \in C^0(\Gamma)$, we now define the generalized SC operator $\mathcal{I}_L^{m,g}$ by

$$\begin{aligned} \mathcal{I}_L^{m,g}[v](\mathbf{y}) &= \sum_{g(\mathbf{l}) \leq L} \left(\Delta^{m(l_1)} \otimes \dots \otimes \Delta^{m(l_N)} \right) [v](\mathbf{y}) \\ &= \sum_{g(\mathbf{l}) \leq L} \sum_{\mathbf{i} \in \{0,1\}^N} (-1)^{|\mathbf{i}|} \left(\mathcal{U}^{m(l_1-i_1)} \otimes \dots \otimes \mathcal{U}^{m(l_N-i_N)} \right) [v](\mathbf{y}), \end{aligned} \quad (4.2)$$

where $\mathbf{i} = (i_1, \dots, i_N)$ is a multi-index with $i_n \in \{0, 1\}$, $|\mathbf{i}| = i_1 + \dots + i_N$, and $L \in \mathbb{N}_+$ represents the approximation level. This approximation lives in the tensor product polynomial space given by

$$\mathcal{P}_{\Lambda_L^{m,g}} = \text{span} \left\{ \prod_{n=1}^N y_n^{l_n} \mid \mathbf{l} \in \Lambda_L^{m,g} \right\},$$

where the multi-index set is defined as follows

$$\Lambda_L^{m,g} = \left\{ \mathbf{l} \in \mathbb{N}^N \mid g(\mathbf{m}^\dagger(\mathbf{l} + \mathbf{1})) \leq L \right\}.$$

Here $\mathbf{m}(\mathbf{l}) = (m(l_1), \dots, m(l_N))$, and $m^\dagger(l) := \min\{w \in \mathbb{N}_+ : m(w) \geq l\}$ is the left inverse of m (see [2]).

Specific choices for the one-dimensional growth rate $m(l)$ and the function $g(\mathbf{l})$ are needed to define the multi-index set $\Lambda_L^{m,g}$ and the corresponding polynomial space $\mathcal{P}_{\Lambda_L^{m,g}}$ for the approximation. In this work, we construct the interpolant in (4.2) using the anisotropic Smolyak construction, i.e.,

$$m(1) = 1, \quad m(l) = 2^{l-1} + 1 \text{ for } l > 1 \text{ and } g(\mathbf{l}) = \sum_{n=1}^N \frac{\alpha_n}{\alpha_{\min}} (l_n - 1), \quad (4.3)$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}_+^N$ is a vector of weights reflecting the anisotropy of the system, i.e., the relative importance of each dimension, with $\alpha_{\min} := \min_n \alpha_n$ (see [30] for more details). Our analysis does not depend strongly on this choice of m and g , and we could use other functions, e.g., $m(l) = l$ and $g(\mathbf{l}) = \max_n \alpha_n l_n$ define the anisotropic tensor product approximation.

When Γ is a bounded domain in \mathbb{R}^N , a common choice is the Clenshaw-Curtis abscissas [10] given by the sets of extrema of Chebyshev polynomials including the end-point extrema. For a sample set of any size $m(l) > 1$, the abscissas in the standard domain $[-1, 1]$ are given by

$$\vartheta^l = \left\{ y_j^l \in [-1, 1] \mid y_j^l = -\cos\left(\frac{\pi(j-1)}{m(l)-1}\right) \text{ for } j = 1, \dots, m(l) \right\}. \quad (4.4)$$

By taking $y_1^1 = 0$ and letting $m(l)$ grow according to the rule in (4.3), one gets a sequence of nested sets $\vartheta^l \subset \vartheta^{l+1}$ for $l \in \mathbb{N}_+$. In addition, with $g(l)$ defined as in (4.3), the resulting set of N -dimensional abscissas is a Clenshaw-Curtis sparse grid. Other nested families of sparse grids can be constructed from, e.g., the Leja points [13], Gauss-Patterson [42], etc.

REMARK 4.1. (Specific Choice of m, g). *For the remainder of the paper, we will assume that the functions m and g are given as in (4.3), and use an underlying Clenshaw-Curtis sparse grid. For simplicity, we will also only consider isotropic collocation methods, i.e. $\alpha_1 = \alpha_2 = \dots = \alpha_N$. We then lighten the notation by defining $\mathcal{I}_L := \mathcal{I}_L^{m,g}$.*

Construction of the approximation $\mathcal{I}_L[v] := \mathcal{I}_L^{m,g}[v]$ requires evaluation of v on a set of collocation points $\mathcal{H}_L \subset \Gamma$ with cardinality M_L . In our case, since the one-dimensional point sets are nested, i.e., $\vartheta^l \subset \vartheta^{l+1}$ for $l \in \mathbb{N}_+$, so that the multi-dimensional point set used by $\mathcal{I}_L[v]$ is given by

$$\mathcal{H}_L = \bigcup_{g(\mathbf{l})=L} (\vartheta^{l_1} \otimes \dots \otimes \vartheta^{l_N}),$$

and the nested structure is preserved, i.e., $\mathcal{H}_L \subset \mathcal{H}_{L+1}$, to satisfy assumption 3.1. Define the difference of the sets $\Delta\mathcal{H}_L := \mathcal{H}_L \setminus \mathcal{H}_{L-1}$, and the number of new collocation points $\Delta M_L = \#(\Delta\mathcal{H}_L)$. With this nestedness condition, the approximation $\mathcal{I}_L[v]$ is a Lagrange interpolating polynomial [31], and thus (4.2) can be rewritten as a linear combination of Lagrange basis functions,

$$\begin{aligned} \mathcal{I}_L[v](\mathbf{y}) &= \sum_{j=1}^{M_L} v(\mathbf{y}_{L,j}) \Psi_{L,j}(\mathbf{y}) \\ &= \sum_{j=1}^{M_L} v(\mathbf{y}_{L,j}) \underbrace{\sum_{\mathbf{l} \in \mathcal{J}(L,j)} \sum_{\mathbf{i} \in \{0,1\}^N} (-1)^{|\mathbf{i}|} \prod_{n=1}^N \psi_{k_n(j)}^{l_n - i_n}(y_n)}_{\Psi_{L,j}(\mathbf{y})}, \end{aligned} \quad (4.5)$$

where the index set $\mathcal{J}(L, j)$ is defined by

$$\mathcal{J}(L, j) = \left\{ \mathbf{l} \in \mathbb{N}_+^N \left| g(\mathbf{l}) \leq L \text{ and } \mathbf{y}_{L,j} \in \bigotimes_{n=1}^N \vartheta^{l_n - i_n} \text{ with } \mathbf{i} \in \{0,1\}^N \right. \right\}.$$

For a given L and j , this represents the subset of multi-indices corresponding to the tensor-product operators $\mathcal{U}^{m(l_1 - i_1)} \otimes \dots \otimes \mathcal{U}^{m(l_N - i_N)}$ in (4.2) with the supporting point $\mathbf{y}_{L,j}$. Then for each $\mathbf{l} \in \mathcal{J}(L, j)$ and $\mathbf{i} \in \{0,1\}^N$, the function $\prod_{n=1}^N \psi_{k_n(j)}^{l_n - i_n}(y_n)$ with $k_n(j) \in \{1, \dots, m(l_n - i_n)\}$, $n = 1, \dots, N$, represents the unique Lagrange basis function for the operator $\mathcal{U}^{m(l_1 - i_1)} \otimes \dots \otimes \mathcal{U}^{m(l_N - i_N)}$ corresponding to $\mathbf{y}_{L,j}$. Therefore, the functions $\{\Psi_{L,j}\}_{j=1}^{M_L}$ are given by a linear combination of tensorized Lagrange polynomials satisfying the “delta property”, i.e., $\Psi_{L,j'}(\mathbf{y}_{L,j}) = \delta_{jj'}$ for $j, j' = 1, \dots, M_L$, and is in the required form of (3.3).

Finally, to construct the fully-discrete approximation in the space $W_h(D) \otimes \mathcal{P}_{\Lambda_L^{m,g}}(\Gamma)$ we apply the interpolation operator $\mathcal{I}_L[\cdot]$, given by (4.5), to the semi-discrete solution $u_h(x, \mathbf{y})$ in (3.1) to obtain:

$$u_{h,L}(x, \mathbf{y}) = \mathcal{I}_L[u_h](x, \mathbf{y}) = \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x) \right) \Psi_{L,j}(\mathbf{y}). \quad (4.6)$$

Due to the delta property of the basis function $\Psi_{L,j}(\mathbf{y})$, the interpolation matrix for $\mathcal{I}_L[u_h]$ is a diagonal matrix, and thus the coefficient vectors $\mathbf{c}_{L,j} = (c_{L,j,1}, \dots, c_{L,j,M_h})$ for $j = 1, \dots, M_L$ can be computed by *independently* solving M_L systems of type (3.2).

4.1. Error estimates for fixed L . In what follows, we focus on the linear elliptic problem (2.2) described in Examples 2.1 and 3.2, and present a detailed convergence and complexity analysis of a fully discrete SC approximation, denoted $\tilde{u}_{h,L}$, for any fixed level, $1 \leq L \leq L_{\max}$. As specified in Remark 4.1, in this section we consider only the isotropic Smolyak version of SC interpolant given by (4.2), defined on Clenshaw-Curtis abscissas, for solving the parameterized linear elliptic PDE. However, our analysis can be extended without any essential difficulty to anisotropic SC methods and more complicated underlying PDEs.

The parameterized elliptic PDE (2.2) admits a weak form that is a symmetric, uniformly coercive and continuous bilinear operator on $H_0^1(D)$; i.e., there exist $\alpha, \beta > 0$, depending on a_{\min} and a_{\max} but independent of \mathbf{y} , such that for every $v, w \in H_0^1(D)$,

$$\left| \int_D a(\mathbf{y}) \nabla v \nabla w \, dx \right| \leq \alpha \|v\|_{H_0^1(D)} \|w\|_{H_0^1(D)} \quad \text{and} \quad \beta \|v\|_{H_0^1(D)}^2 \leq \int_D a(\mathbf{y}) |\nabla v|^2 \, dx.$$

In this case, the bilinear form induces a norm $\|v\|^2 = \int_D a(\mathbf{y}) |\nabla v|^2 \, dx$, which for functions $v(x) = \sum_{i=1}^{M_h} c_i \phi_i(x) \in W_h(D)$, with $\mathbf{c} = (c_1, \dots, c_{M_h})$, coincides with the discrete norm $\|\mathbf{c}\|_{\mathbf{A}(\mathbf{y})}$, where the matrix $\mathbf{A}(\mathbf{y})$ is defined in (3.6). Thus we have

$$\text{Continuity:} \quad \|\mathbf{c}\|_{\mathbf{A}(\mathbf{y})} = \|v\| \leq \sqrt{\alpha} \|v\|_{H_0^1(D)}, \quad \text{and}, \quad (4.7a)$$

$$\text{Ellipticity:} \quad \sqrt{\beta} \|v\|_{H_0^1(D)} \leq \|v\| = \|\mathbf{c}\|_{\mathbf{A}(\mathbf{y})}. \quad (4.7b)$$

We next state some regularity conditions on the parameterized solution $u : \Gamma \rightarrow H_0^1(D)$ to the parameterized elliptic PDE in Examples 2.1 and 3.2.

ASSUMPTION 4.1. (Polyellipse analyticity). *Let $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N) \in (1, \infty)^N$, and assume that $u : \Gamma \rightarrow H_0^1(D)$ admits a complex extension $u^* : \mathbb{C}^N \rightarrow H_0^1(D)$, which is analytic on the polyellipse*

$$\Sigma(\boldsymbol{\gamma}) = \prod_{1 \leq n \leq N} \Sigma(n; \gamma_n) \subset \mathbb{C}^N,$$

where $\Sigma(n; \gamma_n)$ denotes the region bounded by the Bernstein ellipse,

$$\Sigma(n; \gamma_n) = \left\{ \frac{1}{2} (z_n + z_n^{-1}) : z_n \in \mathbb{C}, |z_n| \leq \gamma_n \right\}.$$

The set $\Sigma(\boldsymbol{\gamma}) \subset \mathbb{C}^N$ is the product of ellipses in the complex plane, with foci $z_n = \pm 1$, which are the endpoints of the domain $\Gamma_n, n = 1, \dots, N$. Such ellipses are common in proving convergence results for global interpolation schemes. Conditions under which u satisfies Assumption 4.1 can be found in [12, Theorem 1.2] and [14, Theorem 2.5].

In order to investigate the complexity of the fully discrete approximation $\tilde{u}_{h,L}, L \in \mathbb{N}_+$, we first need to derive sufficient conditions for the error $\|u - \tilde{u}_{h,L}\|_{L_\rho^2}$ to achieve a tolerance of $\varepsilon > 0$, where $L_\rho^2 := L_\rho^2(\Gamma; H_0^1(D))$. Using the triangle inequality, the total error can be split into three parts, i.e.,

$$\|u - \tilde{u}_{h,L}\|_{L_\rho^2} \leq \underbrace{\|u - u_h\|_{L_\rho^2}}_{e_1} + \underbrace{\|u_h - u_{h,L}\|_{L_\rho^2}}_{e_2} + \underbrace{\|u_{h,L} - \tilde{u}_{h,L}\|_{L_\rho^2}}_{e_3}. \quad (4.8)$$

The contributions of e_1 and e_2 correspond to the FEM and SC errors, respectively, and have been previously examined [31]. The error e_3 contributed by the linear solver is often omitted from the analysis in the literature, and in practice can be controlled by setting a tight tolerance on the iterative solver. However, the analysis presented here is focused on providing cost estimates for the iterative solver and requires careful consideration of this term. First, we recall error estimates for e_1 and e_2 , given from [31].

LEMMA 4.1. *Let \mathcal{T}_h be a uniform finite element mesh over $D \subset \mathbb{R}^d, d = 1, 2, 3$, with $M_h = \mathcal{O}(1/h^d)$ grid points. For the random elliptic PDE in Example (2.1), when $u(x, \mathbf{y}) \in L^2_\varrho(\Gamma; H^1_0(D) \cap H^{s+1}(D)), s \in \mathbb{N}_+$, the error of the finite element approximation u_h is bounded by*

$$\|u - u_h\|_{L^2_\varrho} \leq C_{\text{fem}} h^s, \quad (4.9)$$

where the constant C_{fem} is independent of h and \mathbf{y} .

LEMMA 4.2. *Let u satisfy Assumption 4.1. For $L \in \mathbb{N}^+$, the interpolation error $u - \mathcal{I}_L[u]$ of the sparse grid SC method using Clenshaw-Curtis abscissas can be bounded as*

$$\|u - \mathcal{I}_L[u]\|_{L^\infty(\Gamma; H^1_0(D))} \leq C_{\text{sc}} e^{-rN2^{L/N}}, \quad (4.10)$$

where, for a constant $0 < \delta < 1$, the rate $r = (1 - \delta) \min_{1 \leq n \leq N} \log \gamma_n$, and the constant $C_{\text{sc}} > 0$ depends on N , u , and δ . We remark that the projection of u into the finite element subspace, denoted u_h , also satisfies Assumption 4.1 with the same region of analyticity, and therefore the application of the interpolant, \mathcal{I}_L , to the semidiscrete solution u_h will converge as in (4.10).

We now turn our attention to the global solver error e_3 in (4.8), which is the error incurred from approximating the solution to (3.7) at each sample point. The difference $u_{h,L} - \tilde{u}_{h,L}$ can be written as an interpolant of the solver error, i.e.,

$$u_{h,L} - \tilde{u}_{h,L} = \mathcal{I}_L[u_h - \tilde{u}_h],$$

which represents the solver error amplified by the interpolation operator. For the operator $\mathcal{I}_L[\cdot]$ in (4.5), we have

$$\|u_{h,L} - \tilde{u}_{h,L}\|_{L^\infty(\Gamma; H^1_0(D))} \leq C_L \max_{j=1, \dots, M_L} \|u_h(\mathbf{y}_{L,j}) - \tilde{u}_h(\mathbf{y}_{L,j})\|_{H^1_0(D)}.$$

Thus, from the ellipticity condition in (4.7b),

$$e_3 \leq C_L \max_{j=1, \dots, M_L} \|u_h(\mathbf{y}_{L,j}) - \tilde{u}_h(\mathbf{y}_{L,j})\|_{H^1_0(D)} \leq C_L \frac{1}{\sqrt{\beta}} \max_{j=1, \dots, M_L} \|\mathbf{c}_{L,j} - \tilde{\mathbf{c}}_{L,j}\|_{\mathbf{A}(\mathbf{y}_{L,j})} \leq \frac{\tau}{\sqrt{\beta}} C_L,$$

where τ is defined to be the tolerance of the linear solver. Note that the expression $u_h - \tilde{u}_h$ is only defined at collocation points. The solver error for each fixed $\mathbf{y}_{L,j} \in \mathcal{H}_L$ is controlled by the CG convergence estimate (3.8). The Lebesgue constant of the operator $\mathcal{I}_L[\cdot]$ is defined by $C_L = \max_{\mathbf{y} \in \Gamma} \sum_{j=1}^{M_L} |\Psi_{L,j}(\mathbf{y})|$ where $\Psi_{L,j}$ is given in (4.5). We now provide an upper bound of C_L in the following lemma.

LEMMA 4.3. *The Lebesgue constant for the isotropic sparse-grid interpolation operator $\mathcal{I}_L[\cdot]$ in (4.5) using the Clenshaw-Curtis rule on $\Gamma = \prod_{n=1}^N \Gamma_n = [-1, 1]^N$ is bounded by*

$$C_L \leq [(L+1)(L+2)]^N, \quad (4.11)$$

where L and N are the level of the interpolation operator and dimension of the parameter space,

respectively.

Proof. For each $n = 1, \dots, N$, recall that the Lebesgue constants λ_{l_n} of the one-dimensional operators $\mathcal{U}^{m(l_n)}$ are given by [43]

$$\lambda_{l_n} = \max_{z \in \Gamma_n} \sum_{j=1}^{m(l_n)} \left| \psi_j^{l_n}(z) \right|.$$

For Lagrange interpolants based on Clenshaw-Curtis abscissas (4.4), we have [15]

$$\lambda_{l_n} \leq \frac{2}{\pi} \log(m(l_n) - 1) + 1 \quad \text{for } l_n \geq 2.$$

Combining this with the growth rate $m(l_n) = 2^{l_n-1} + 1$ for $l_n \geq 2$ given by (4.3), it is easy to obtain that

$$\lambda_{l_n} \leq 2l_n - 1 \quad \text{for } l_n \geq 2.$$

For $v \in C^0(\Gamma_n)$, the difference operator $\Delta^{m(l_n)}$ for $l_n = 1$ satisfies

$$\left\| \Delta^{m(1)}[v] \right\|_{L^\infty(\Gamma_n)} = \left\| \mathcal{U}^{m(1)}[v] \right\|_{L^\infty(\Gamma_n)} \leq \lambda_1 \max_{y_n \in \vartheta^1} |v(y_n)|.$$

For $l_n \geq 2$, the triangle inequality yields

$$\begin{aligned} \left\| \Delta^{m(l_n)}[v] \right\|_{L^\infty(\Gamma_n)} &= \left\| \mathcal{U}^{m(l_n)}[v] - \mathcal{U}^{m(l_n-1)}[v] \right\|_{L^\infty(\Gamma_n)} \\ &\leq (\lambda_{l_n} + \lambda_{l_n-1}) \max_{y_n \in \vartheta^{l_n}} |v(y_n)|. \end{aligned}$$

Finally, for $v \in C^0(\Gamma)$, we bound the interpolant $\mathcal{I}_L[v]$ by

$$\begin{aligned} \left\| \mathcal{I}_L[v] \right\|_{L^\infty(\Gamma)} &= \left\| \sum_{g(\mathbf{l}) \leq L} \Delta^{m(l_1)} \otimes \dots \otimes \Delta^{m(l_N)}[v] \right\|_{L^\infty(\Gamma)} \\ &\leq \left(2^N \sum_{g(\mathbf{l}) \leq L} \prod_{n=1}^N l_n \right) \max_{j=1, \dots, M_L} |v(\mathbf{y}_{L,j})| \leq 2^N \left(\sum_{l=1}^{L+1} l \right)^N \max_{j=1, \dots, M_L} |v(\mathbf{y}_{L,j})| \\ &= [(L+1)(L+2)]^N \max_{j=1, \dots, M_L} |v(\mathbf{y}_{L,j})|, \end{aligned}$$

which gives the desired estimate. \square

4.2. Complexity analysis. Now we analyze the cost of constructing $\tilde{u}_{h, L_{\max}}, L_{\max} \in \mathbb{N}_+$, with the prescribed accuracy ε . Here we assume $\varepsilon > 0$ is sufficiently small, and study the asymptotic growth of the total costs (3.10) for the accelerated construction of $\tilde{u}_{h, L_{\max}}$, described in §3. For comparison, we will also analyze the cost (3.9) associated with the standard SC method, where iterative solvers for the sequence of solutions to the linear systems (3.7) are seeded with the zero vector as an initial guess. According to the error estimates discussed in §4.1, a sufficient condition

to ensure $\|u - \tilde{u}_{h, L_{\max}}\|_{L_\theta^2} \leq \varepsilon$ is that

$$\|e_1\|_{L_\theta^2} \leq C_{\text{fem}} h^s \leq \frac{\varepsilon}{3}, \quad (4.12a)$$

$$\|e_2\|_{L_\theta^2} \leq \|e_2\|_{L_\theta^\infty} \leq C_{\text{sc}} e^{-rN2^{L_{\max}/N}} \leq \frac{\varepsilon}{3}, \quad (4.12b)$$

$$\|e_3\|_{L_\theta^2} \leq \|e_3\|_{L_\theta^\infty} \leq (L_{\max} + 2)^{2N} \frac{\tau}{\sqrt{\beta}} \leq \frac{\varepsilon}{3}. \quad (4.12c)$$

In section §3 we defined K_{zero} and K_{acc} as the total number of solver iterations used by the standard and accelerated SC methods, respectively, to solve (3.7) at each sample point. Now let $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$ represent the minimum values of K_{zero} and K_{acc} , respectively, needed to satisfy the inequalities (4.12). Here we aim to estimate upper bounds of $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$. Note that, for fixed dimension N , level L_{\max} , and mesh size h , the total number of iterations is determined by the inequality (4.12c). Larger values of L_{\max} and $1/h$, lead to higher costs. Thus, the estimation of $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$ has two steps: (i) Given N and ε , estimate the maximum possible h to satisfy (4.12a) and the minimum L_{\max} that achieves (4.12b); (ii) Substitute the obtained values into (4.12c) to estimate upper bounds on $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$ according to the CG error estimate (3.8). For (i), we have the following lemma, that follows immediately from Lemmas 4.1 and 4.2.

LEMMA 4.4. *Given the assumptions of Lemmas 4.1 and 4.2, the error bounds (4.12a) and (4.12b) can be achieved by choosing finite element mesh size h and the sparse-grid level L_{\max} according to*

$$h(\varepsilon) = \left(\frac{\varepsilon}{3C_{\text{fem}}} \right)^{1/s} \quad \text{and} \quad L_{\max}(\varepsilon) = \left\lceil \frac{N}{\log 2} \log \left(\frac{1}{rN} \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right) \right\rceil. \quad (4.13)$$

For convenience, we treat the integer quantities $K_{\text{zero}}(\varepsilon)$, $K_{\text{acc}}(\varepsilon)$, and $L_{\max}(\varepsilon)$ as positive real numbers in the rest of this section. Now, based on the estimate in Lemma 4.3 for the Lebesgue constant $C_{L_{\max}}$, we state the following lemma related to the choice of an appropriate tolerance $\tau(\varepsilon)$ to satisfy the error bounds (4.12c).

LEMMA 4.5. *Let $\varepsilon > 0$. Given the assumptions of Lemmas 4.1 and 4.2, a sufficient condition to ensure $e_3 < \varepsilon/3$ is that*

$$\tau(\varepsilon) = \frac{\sqrt{\beta} \varepsilon}{3(L_{\max}(\varepsilon) + 2)^{2N}}. \quad (4.14)$$

Moreover, it holds

$$\frac{1}{\sqrt{\beta}} (L + 2)^{2N} \tau(\varepsilon) \leq C_{\text{sc}} e^{-rN2^{L/N}} \quad \text{for } L = 0, \dots, L_{\max}(\varepsilon) - 1,$$

where $L_{\max}(\varepsilon)$ is the minimum level given in (4.13).

Proof. It is easy to see that (4.14) is an immediate result of (4.12c). For $L = 0, \dots, L_{\max}(\varepsilon) - 1$, we have

$$\frac{1}{\sqrt{\beta}} (L + 2)^{2N} \tau(\varepsilon) \leq \frac{1}{\sqrt{\beta}} (L_{\max}(\varepsilon) + 2)^{2N} \tau(\varepsilon) \leq \frac{\varepsilon}{3} \leq C_{\text{sg}} e^{-rN2^{(L_{\max}(\varepsilon)-1)/N}} \leq C_{\text{sg}} e^{-rN2^{L/N}},$$

which completes the proof. \square

Using the selected $h := h(\varepsilon)$, $L_{\max} := L_{\max}(\varepsilon)$, and $\tau := \tau(\varepsilon)$, we now estimate the upper

bounds on the number of CG iterations needed to solve a linear system at a point $\mathbf{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}$. To proceed, define

$$k_{\text{zero}} := \max_{\mathbf{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}} k_{L_{\max},j} \quad \text{and} \quad k_{\text{acc}}^L := \max_{\mathbf{y}_{L,j} \in \Delta \mathcal{H}_L} k_{L,j} \quad \text{for } L = 1, \dots, L_{\max},$$

where $k_{L,j}$ is the number of CG iterations required to achieve $\|\mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(k_{L,j})}\|_{\mathbf{A}_{L,j}} \leq \tau(\varepsilon)$, which, in general, depends on the choice of initial vector. Note that, in the case $\mathbf{c}_{L,j}^{(0)} = (0, \dots, 0)^\top$, there is no improvement in the iteration count as the level L increases, so k_{zero} does not depend on L . Now we give the following estimates on k_{zero} and $\{k_{\text{acc}}^L\}_{L=1}^{L_{\max}}$.

LEMMA 4.6. *Under the conditions of Lemmas 4.1 and 4.2, for any $\mathbf{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}$, if the CG method with zero initial vector is used to solve (3.7) to tolerance $\tau > 0$, then k_{zero} can be bounded by*

$$k_{\text{zero}} \leq \log \left(\frac{2\sqrt{\alpha} \|u_h\|_{L^\infty(\Gamma; H_0^1(D))}}{\tau} \right) \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}} + 1}{\sqrt{\bar{\kappa}} - 1} \right). \quad (4.15)$$

Here $\bar{\kappa} = \sup_{\mathbf{y} \in \Gamma} \kappa(\mathbf{y})$, with $\kappa(\mathbf{y})$ the condition number of the matrix $\mathbf{A}(\mathbf{y})$ corresponding to (3.2). Alternatively, if the initial vector is given by the acceleration method as in (3.5), then k_{acc}^L can be bounded by

$$k_{\text{acc}}^L \leq \log \left(\frac{4\sqrt{\alpha} C_{\text{sc}} e^{-rN2^{(L-1)/N}}}{\tau} \right) \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}} + 1}{\sqrt{\bar{\kappa}} - 1} \right), \quad (4.16)$$

for $L = 1, \dots, L_{\max}$.

Proof. Let $\mathbf{y}_{L,j}$ be an arbitrary point in \mathcal{H}_L , $1 \leq L \leq L_{\max}$. Given an initial guess $\mathbf{c}_{L,j}^{(0)}$, the minimum number of CG iterations needed to achieve tolerance $\tau > 0$ can be obtained immediately from (3.8), that is,

$$k_{L,j} = \left\lceil \log \left(\frac{2\|\mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(0)}\|_{\mathbf{A}_{L,j}}}{\tau} \right) \bigg/ \log \left(\frac{\sqrt{\kappa_{L,j}} + 1}{\sqrt{\kappa_{L,j}} - 1} \right) \right\rceil,$$

where $\mathbf{A}_{L,j} = \mathbf{A}(\mathbf{y}_{L,j})$ is the FE system matrix corresponding to parameter $\mathbf{y}_{L,j}$, and $\kappa_{L,j} = \kappa(\mathbf{y}_{L,j})$ is the condition number of $\mathbf{A}_{L,j}$ (See Example 3.2). In the case that $\mathbf{c}_{L,j}^{(0)} = (0, \dots, 0)^\top$, the estimate in (4.15) can be obtained from (4.7a), i.e.,

$$\|\mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(0)}\|_{\mathbf{A}_{L,j}} = \|\mathbf{c}_{L,j}\|_{\mathbf{A}_{L,j}} \leq \sqrt{\alpha} \|u_h\|_{L^\infty(\Gamma; H_0^1(D))}.$$

Alternatively, when using $\tilde{u}_{h,L-1}$ for $L = 1, \dots, L_{\max}$ to provide initial vectors for the CG solver (based on (3.5)), for $\mathbf{y}_{L,j} \in \Delta \mathcal{H}_L$ we use Lemma 4.5 and (4.7a) to get the following estimate:

$$\begin{aligned} \|\mathbf{c}_{L,j} - \mathbf{c}_{L,j}^{(0)}\|_{\mathbf{A}_{L,j}} &\leq \sqrt{\alpha} \|u_h - \tilde{u}_{h,L-1}\|_{L^\infty(\Gamma; H_0^1(D))} \\ &\leq \sqrt{\alpha} \left(\|u_h - u_{h,L-1}\|_{L^\infty(\Gamma; H_0^1(D))} + \|u_{h,L-1} - \tilde{u}_{h,L-1}\|_{L^\infty(\Gamma; H_0^1(D))} \right) \\ &\leq \sqrt{\alpha} \left(C_{\text{sc}} e^{-rN2^{(L-1)/N}} + \frac{1}{\sqrt{\beta}} (L+1)^{2N} \tau \right) \\ &\leq 2\sqrt{\alpha} C_{\text{sc}} e^{-rN2^{(L-1)/N}}. \end{aligned}$$

This leads directly to the estimate in (4.16). \square

In the accelerated case, the sparse-grid interpolant $\mathcal{I}_{L_{\max}}[u_h]$ must be constructed in the following fashion: before solving the system (3.7) corresponding to a sample point $\mathbf{y}_{L,j} \in \Delta\mathcal{H}_L$, we must first solve the systems for all sample points in \mathcal{H}_{L-1} . With a total number $\Delta M_L = \#(\Delta\mathcal{H}_L)$ of new linear systems at level L , the total number of CG iterations for the newly added points at level L can be bounded by $\Delta M_L k_{\text{zero}}$ and $\Delta M_L k_{\text{acc}}^L$, for the standard and the accelerated cases, respectively. Then since $M_{L_{\max}} = \sum_{L=1}^{L_{\max}} \Delta M_L$, we find that the total number of iterations for the standard and accelerated schemes can be bounded as

$$K_{\text{zero}}(\varepsilon) \leq M_{L_{\max}} k_{\text{zero}}, \quad \text{and} \quad K_{\text{acc}}(\varepsilon) \leq \sum_{L=1}^{L_{\max}} \Delta M_L k_{\text{acc}}^L.$$

This leads to the following estimates.

THEOREM 4.7. *Given Assumption 4.1, and the conditions of Lemmas 4.1 and 4.2, for $\varepsilon > 0$, the minimum total number of CG iterations $K_{\text{zero}}(\varepsilon)$ to achieve $\|u - \tilde{u}_{h,L_{\max}}\|_{L_\theta^2} < \varepsilon$, using zero initial vectors is bounded by*

$$\begin{aligned} K_{\text{zero}}(\varepsilon) \leq & C_1 \left[\log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right]^N \left[C_2 + \frac{1}{\log 2} \log \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right]^{N-1} \\ & \times \frac{1}{\log \left(\frac{\sqrt{\bar{\kappa}}+1}{\sqrt{\bar{\kappa}}-1} \right)} \left\{ \log \left(\frac{C_3}{\varepsilon} \right) + C_4 + 2N \log \log \left[\frac{1}{rN} \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right] \right\}, \end{aligned} \quad (4.17)$$

where $\bar{\kappa}$ is as defined in Lemma 4.6, and the constants C_1 , C_2 , C_3 and C_4 are defined by

$$\begin{aligned} C_1 &= \left(\frac{e}{\log 2} \right)^{N-1} \left(\frac{2}{rN} \right)^N, \quad C_2 = 1 + \frac{1}{\log 2} \log \left(\frac{1}{rN} \right), \\ C_3 &= 6\sqrt{\frac{\alpha}{\beta}} \|u_h\|_{L^\infty(\Gamma; H_0^1(D))}, \quad C_4 = 2N \log \left(\frac{2N}{\log 2} \right). \end{aligned} \quad (4.18)$$

Proof. To achieve the prescribed error, we balance the three error sources that contribute to the total error (4.8). To control e_1 and e_2 , set $h = h(\varepsilon)$ and $L_{\max} = L_{\max}(\varepsilon)$ according to Lemma 4.4. For the solver error e_3 , we choose the solver tolerance $\tau = \tau(\varepsilon)$ according to Lemma 4.5. Then, the total number of iterations $K_{\text{zero}}(\varepsilon)$ can be bounded by

$$K_{\text{zero}}(\varepsilon) = \sum_{j=1}^{M_{L_{\max}}} \leq M_{L_{\max}} k_{\text{zero}}. \quad (4.19)$$

From Lemma 4.5 and 4.6, we have

$$\begin{aligned} k_{\text{zero}} &\leq \log \left(\frac{2\sqrt{\alpha} \|u_h\|_{L^\infty(\Gamma; H_0^1(D))}}{\tau} \right) \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}}+1}{\sqrt{\bar{\kappa}}-1} \right) \\ &\leq \log \left(\frac{6\sqrt{\alpha} \|u_h\|_{L^\infty(\Gamma; H_0^1(D))} (L_{\max} + 2)^{2N}}{\sqrt{\beta}\varepsilon} \right) \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}}+1}{\sqrt{\bar{\kappa}}-1} \right) \end{aligned} \quad (4.20)$$

$$\begin{aligned}
&\leq \left[\log \left(\frac{C_3}{\varepsilon} \right) + 2N \log (L_{\max} + 2) \right] \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}} + 1}{\sqrt{\bar{\kappa}} - 1} \right) \\
&\leq \left\{ \log \left(\frac{C_3}{\varepsilon} \right) + C_4 + 2N \log \log \left(\frac{1}{rN} \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right) \right\} \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}} + 1}{\sqrt{\bar{\kappa}} - 1} \right).
\end{aligned}$$

In addition, following [31, Lemma 3.9], we bound the number of interpolation points:

$$\begin{aligned}
M_{L_{\max}} &\leq \sum_{L=1}^{L_{\max}} 2^L \binom{N-1+L}{N-1} \leq \sum_{L=1}^{L_{\max}} 2^L \left(1 + \frac{L}{N-1} \right)^{N-1} e^{N-1} \\
&\leq e^{N-1} 2^{L_{\max}+1} \left(1 + \frac{L_{\max}}{N-1} \right)^{N-1} \\
&\leq 2e^{N-1} \left\{ \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right\}^N \left\{ C_2 + \frac{1}{\log 2} \log \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right\}^{N-1},
\end{aligned} \tag{4.21}$$

where in the last line we have used (4.13) to replace L_{\max} . Substituting (4.20) and (4.21) into (4.19) concludes the proof. \square

THEOREM 4.8. *Given Assumption 4.1, and the conditions of Lemmas 4.1 and 4.2, for $\varepsilon > 0$, the minimum total number of CG iterations $K_{\text{acc}}(\varepsilon)$, to achieve $\|u - \hat{u}_{h, L_{\max}}\|_{L^2_\varepsilon} < \varepsilon$, in Algorithm 1, is bounded by*

$$\begin{aligned}
K_{\text{acc}}(\varepsilon) &\leq C_1 \left[\log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right]^N \left[C_2 + \frac{1}{\log 2} \log \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right]^{N-1} \\
&\quad \times \frac{1}{\log \left(\frac{\sqrt{\bar{\kappa}}+1}{\sqrt{\bar{\kappa}}-1} \right)} \left\{ C_5 + 2 \left(2^{\frac{1}{N}} - 1 \right) \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) + 2N \log \log \left[\frac{1}{rN} \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right] \right\},
\end{aligned} \tag{4.22}$$

where $\bar{\kappa} = \sup_{\mathbf{y} \in \Gamma} (\kappa(\mathbf{y}))$ as in Lemma 4.6, C_1 and C_2 are defined as in (4.18), and C_5 is defined by

$$C_5 = 2N \log \left(\frac{2N}{\log 2} \right) + \log \left(4 \sqrt{\frac{\alpha}{\beta}} \right).$$

Proof. To achieve the prescribed error, we again choose $h = h(\varepsilon)$, $L_{\max} = L_{\max}(\varepsilon)$ and $\tau = \tau(\varepsilon)$ as in Lemmas 4.4 and 4.5. Then, the total number of iterations $K_{\text{acc}}(\varepsilon)$ can be bounded by

$$K_{\text{acc}}(\varepsilon) = \sum_{L=1}^{L_{\max}} \sum_{\mathbf{y}_{L,j} \in \Delta \mathcal{H}_L} k_{L,j} \leq \sum_{L=1}^{L_{\max}} \Delta M_L k_{\text{acc}}^L.$$

From Lemma 4.5 and 4.6, for $L = 1, \dots, L_{\max}$, we have

$$\begin{aligned}
k_{\text{acc}}^L &\leq \log \left(\frac{4\sqrt{\alpha}C_{\text{sc}} e^{-rN2^{(L-1)/N}}}{\tau} \right) \bigg/ \log \left(\frac{\sqrt{\bar{\kappa}} + 1}{\sqrt{\bar{\kappa}} - 1} \right) \\
&\leq \frac{1}{\log \left(\frac{\sqrt{\bar{\kappa}}+1}{\sqrt{\bar{\kappa}}-1} \right)} \log \left(\frac{12\sqrt{\alpha}C_{\text{sc}}C_{L_{\max}} e^{-rN2^{(L-1)/N}}}{\sqrt{\beta}\varepsilon} \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \log \left[\left(\frac{3C_{\text{sc}} e^{-rN2^{L/N}}}{\varepsilon} \right) 4\sqrt{\frac{\alpha}{\beta}} C_{L_{\max}} e^{rN2^{L/N} - rN2^{(L-1)/N}} \right] \\
&\leq \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \log \left(4\sqrt{\frac{\alpha}{\beta}} C_{L_{\max}} e^{rN(2^{L/N} - 2^{(L-1)/N})} \right) \\
&= \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \left[\log \left(4\sqrt{\frac{\alpha}{\beta}} C_{L_{\max}} \right) + rN \left(2^{L/N} - 2^{(L-1)/N} \right) \right].
\end{aligned}$$

Hence,

$$K_{\text{acc}}(\varepsilon) \leq M_{L_{\max}} \frac{\log \left(4\sqrt{\alpha/\beta} C_{L_{\max}} \right)}{\log \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)} + \frac{rN}{\log \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)} \underbrace{\sum_{L=1}^{L_{\max}} \Delta M_L \left(2^{L_{\max}/N} - 2^{(L-1)/N} \right)}_S,$$

where S can be bounded using results from geometric sums, i.e.,

$$\begin{aligned}
S &\leq \sum_{L=1}^{L_{\max}} 2^L \binom{N-1+L}{N-1} \left(2^{L_{\max}/N} - 2^{(L-1)/N} \right) \\
&\leq e^{N-1} \left(1 + \frac{L_{\max}}{N-1} \right)^{N-1} \sum_{L=1}^{L_{\max}} \left(2^{L_{\max}/N} - 2^{(L-1)/N} \right) 2^L \\
&= e^{N-1} \left(1 + \frac{L_{\max}}{N-1} \right)^{N-1} \left\{ \left(1 - \frac{1}{2^{1+1/N}} \right) 2^{L_{\max}+1} 2^{L_{\max}/N} + \frac{2}{2^{1+1/N} - 1} - 2^{1+L_{\max}/N} \right\} \\
&\leq e^{N-1} \left(1 + \frac{L_{\max}}{N-1} \right)^{N-1} \left(2^{1/N} - 1 \right) 2^{L_{\max}+2} 2^{L_{\max}/N}.
\end{aligned}$$

Combining the last two inequalities, along with (4.21), we get

$$\begin{aligned}
K_{\text{acc}}(\varepsilon) &\leq e^{N-1} \left(1 + \frac{L_{\max}}{N-1} \right)^{N-1} 2^{L_{\max}+1} \\
&\quad \times \frac{1}{\log \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)} \log \left(4\sqrt{\frac{\alpha}{\beta}} \right) + 2N \log(L_{\max} + 2) + 2rN \left(2^{1/N} - 1 \right) 2^{L_{\max}/N}
\end{aligned}$$

Substituting (4.13) for L_{\max} concludes the proof. \square

In the case of the accelerated SC method, an interpolant $\mathcal{I}_{L-1}[\tilde{u}_h]$, defined by (4.5) and (3.4), must be evaluated for each of the ΔM_L collocation points in $\Delta \mathcal{H}_L$. Each interpolant evaluation costs $2M_{L-1} - 1$ operations, i.e., additions and multiplications, and must be evaluated for each of the M_h components of the FEM coefficient vector. Then the interpolation cost on each level is $M_h \Delta M_L (2M_{L-1} - 1)$ for $L = 1, \dots, L_{\max}(\varepsilon)$. Now we give an estimate of the total interpolation cost $\mathcal{C}_{\text{int}}(\varepsilon)$ for our algorithm to achieve the prescribed accuracy ε .

THEOREM 4.9. *Given Assumption 4.1 and the conditions of Lemma 4.1, for sufficiently small $\varepsilon > 0$, the total cost of interpolation when using the sparse grid interpolation method in (3.5) is*

bounded by

$$\mathcal{C}_{\text{int}}(\varepsilon) \leq M_h C_8 \left(\frac{1}{rN} \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right)^{2N} \left\{ C_2 + \frac{1}{\log 2} \log \log \left(\frac{3C_{\text{sc}}}{\varepsilon} \right) \right\}^{2(N-1)},$$

where C_2 are defined as in Theorem 4.7, and $C_8 = 64 e^{2(N-1)}$.

Proof. The total interpolation cost is bounded by

$$\begin{aligned} \mathcal{C}_{\text{int}}(\varepsilon) &\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} \Delta M_L M_{L-1} \\ &\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} 2^L \binom{N-1+L}{N-1} \sum_{l=1}^L 2^l \binom{N-1+l}{N-1} \\ &\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} 2^L \binom{N-1+L}{N-1} \sum_{l=1}^L 2^l \binom{N-1+l}{N-1} \\ &\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} 2^L \left\{ \binom{N-1+L}{N-1} \right\}^2 2^{L+1} \\ &\leq 4M_h \left\{ \binom{N-1+L_{\max}(\varepsilon)}{N-1} \right\}^2 4^{L_{\max}(\varepsilon)+1} \\ &\leq 16M_h e^{2(N-1)} 4^{L_{\max}(\varepsilon)} \left(1 + \frac{L_{\max}(\varepsilon)}{N-1} \right)^{2(N-1)}. \end{aligned} \tag{4.23}$$

Substituting the definition of $L_{\max}(\varepsilon)$ from Lemma 4.4 into (4.23) concludes the proof. \square

Based on Theorems 4.7, 4.8 and 4.9, we finally discuss the savings of the accelerated SC method proposed in §3. By comparing the estimates of $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$, we see that the acceleration technique reduces $\log(C_3/\varepsilon)$ in (4.17) to $2(2^{1/N} - 1) \log(3C_{\text{sc}}/\varepsilon)$ in (4.22). Here both terms are of the same asymptotic order with respect to ε , but the savings from acceleration increases with dimension N since $(2^{1/N} - 1) \rightarrow 0$ as $N \rightarrow \infty$. On the other hand, when taking into account the cost of interpolation \mathcal{C}_{int} , we must consider the cost $\mathcal{C}_{\text{iter}}$ of performing each iteration. In the case of using CG solvers, $\mathcal{C}_{\text{iter}}$ is the cost of one matrix-vector multiplication, and will be determined by the size of the unknown vector, M_h , and the sparsity of the mass matrix $\mathbf{A}(\mathbf{y})$. Thus $\mathcal{C}_{\text{iter}}$ is proportional to the size of the finite element vector, i.e., $\mathcal{C}_{\text{iter}} = C_D M_h$, where C_D depends on the dimension d of the physical domain and choice of finite element basis. For example, without the use of a preconditioner, we can assume that the condition numbers of the matrices $\mathbf{A}(\mathbf{y})$, for $\mathbf{y} \in \Gamma$, satisfy

$$\bar{\kappa} := \sup_{\mathbf{y} \in \Gamma} \kappa(\mathbf{y}) \leq \left(\frac{C_\kappa}{h} \right)^2,$$

where the constant $C_\kappa > 0$ is independent of $\mathbf{y} \in \Gamma$ [3]. Then we can examine the contribution of the condition number in Theorems 4.7 and 4.8: using the inequality $\log(x) \geq (x-1)/x$ and Lemmas

4.1 and 4.4, we bound the terms involving the condition number as

$$\frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \leq \frac{\sqrt{\kappa}+1}{2} \leq C_\kappa \left(\frac{3C_{\text{fem}}}{\varepsilon}\right)^{1/s}.$$

Now as $\varepsilon \rightarrow 0$, the asymptotic iterative solver costs, $C_{\text{zero}} = C_D M_h K_{\text{zero}}$ are of the order $M_h \left(\frac{1}{\varepsilon}\right)^{1/s} \left\{\log\left(\frac{1}{\varepsilon}\right)\right\}^{N+1} \{\log \log(\cdot)\}$ while in the accelerated case, the estimate for $C_D M_h K_{\text{acc}}$, is of the same order with respect to ε , but with an improvement to the constant of $(2^{1/N} - 1)$. For the accelerated method, the additional interpolation costs C_{int} are of order $M_h \left\{\log\left(\frac{1}{\varepsilon}\right)\right\}^{2N} \left\{\log \log\left(\frac{1}{\varepsilon}\right)\right\}^{2(N-1)}$, which is negligible compared to the iterative solver complexity. It is clear that, asymptotically, the accelerated method leads to a net reduction in computational cost. We remark that for many adaptive interpolation methods, the addition of new points already involves evaluation of the current (coarse) interpolant. In this case, the cost of interpolation can be ignored, and the accelerated method should be used.

5. Numerical examples. The goal of this section is to demonstrate the reduction in computational cost of SC methods using the proposed acceleration technique. In Example 5.1, we first use the accelerated SC method to solve an stochastic elliptic PDE with one spatial dimension, and compute the overall cost and iteration savings gained by acceleration. Example 5.2 considers a similar problem and looks at the number of CG iterations versus the collocation error, comparing the implementation of the method using isotropic and anisotropic sparse grids, and demonstrating the effect of varying stochastic dimension N on the convergence of the individual systems. In addition, as described in Remark 3.3, we extend our acceleration technique to interpolated preconditioners, which also exhibit the convergence improvements of the method. Finally, Example 5.3 applies the accelerated method to iterative solvers for nonlinear parametrized PDEs.

The analysis in section 4.1 consisted of two components: (i) estimates for the reduction in solver iterations from using acceleration, and (ii) interpolation costs. The interpolation costs can be computed exactly for non-adaptive methods, and for adaptive implementations of sparse grid SC the interpolation costs can be ignored. In Example 5.1, all error contributions are balanced, and the total cost is examined, including both solver iterations and interpolation construction. In Examples 5.2 and 5.3 we focus only on the number of iterations of the CG solver.

Example 5.1. We consider the following elliptic stochastic PDE

$$\begin{cases} -\nabla \cdot (a(x, \mathbf{y}) \nabla u(x, \mathbf{y})) &= 10 & \text{in } D \times \Gamma, \\ u(x, \mathbf{y}) &= 0 & \text{on } \partial D \times \Gamma, \end{cases} \quad (5.1)$$

where $D = [0, 1]$, $\mathbf{y} = (y_1, y_2, y_3, y_4)^\top$, $\Gamma_n = [-1, 1]$, $n = 1, \dots, 4$, and the coefficient a is given by:

$$\log(a(x, \mathbf{y}) - 1) = e^{-1/8} (y_1 \cos \pi x + y_2 \sin \pi x + y_3 \cos 2\pi x + y_4 \sin 2\pi x). \quad (5.2)$$

The random variables $\{y_i\}_{i=1}^4$ are independent and identically distributed uniform random variables in $[-1, 1]$. In the one-dimensional physical domain, a finite element discretization using linear elements yields tridiagonal, symmetric positive-definite systems. While this type of system could be solved efficiently by direct methods, nevertheless we use CG solvers to demonstrate the convergence properties of the acceleration method.

Table 5.1 compares the standard and the accelerated SC methods, where the error for each approximate solution, $\tilde{u}_{h, L_{\max}}$, is computed against a highly refined approximate reference solution \tilde{u}_{h^*, L^*} with $h^* = 2^{-14}$, $L^* = 10$. In Figure 5.1 we plot the savings of the accelerated SC method, computed according to the cost metrics (3.9) and (3.10). Since the constants C_{fem} and C_{sc} in Lemma

4.4 are not known *a priori*, to balance the error contributions in (4.12) we use trial and error to determine sufficient values h , L_{\max} , and τ to achieve the desired overall error ε in the L^2_ϱ norm. Especially for the larger systems, i.e., those with a large number of spatial degrees of freedom, significant savings are achieved. The percent savings in the number of iterations versus the cost of interpolation are calculated according to

$$\frac{\mathcal{C}_{\text{zero}} - \mathcal{C}_{\text{acc}}}{\mathcal{C}_{\text{zero}}} = \frac{M_h C_D (K_{\text{zero}} - K_{\text{acc}}) - \mathcal{C}_{\text{int}}}{M_h C_D K_{\text{zero}}},$$

where $C_D = 5$, since the matrices are tridiagonal.

Tot. Err	FE DoFs	SC Pts	CG tol	K_{zero}	K_{acc}	Savings
1×10^{-2}	255	137	1×10^{-3}	28,259	21,123	19.4 %
5×10^{-3}	511	401	5×10^{-3}	173,671	83,884	42.4%
1×10^{-3}	2,047	1,105	1×10^{-4}	2,001,905	626,215	62.3%
5×10^{-4}	4,095	2,929	5×10^{-5}	10,878,352	1,842,703	74.5%
1×10^{-4}	16,383	7,537	1×10^{-5}	114,570,175	12,345,968	75.1%

Table 5.1: Comparison in computational cost between the standard and the accelerated SC methods for solving (5.1)–(5.2).

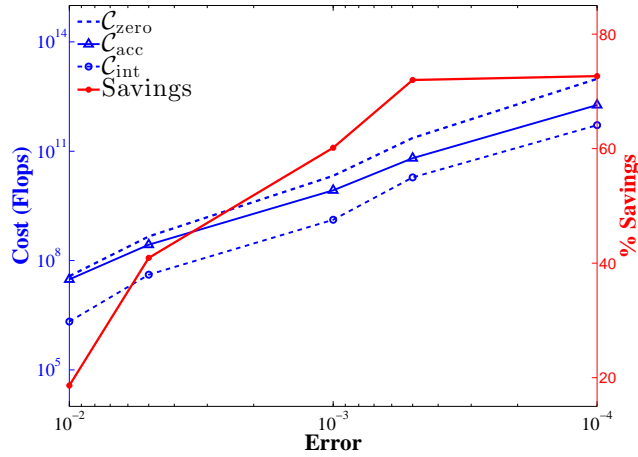


Fig. 5.1: Cost (left axis) and percent savings (right axis) of the accelerated SC method versus the standard SC method for solving (5.1)–(5.2). Costs are computed according to (3.9) and (3.10).

Example 5.2. We consider the following stochastic linear elliptic problem

$$\begin{cases} -\nabla \cdot (a(x, \mathbf{y}) \nabla u(x, \mathbf{y})) = \cos(x_1) \sin(x_2) & \text{in } D \times \Gamma, \\ u(x, \mathbf{y}) = 0 & \text{on } \partial D \times \Gamma, \end{cases} \quad (5.3)$$

where $D = [0, 1] \times [0, 1]$, $\Gamma_n = [-\sqrt{3}, \sqrt{3}]$, $n = 1, \dots, N$, and $x = (x_1, x_2)$ is the spatial variable. The random diffusion term has one-dimensional spatial dependence given by

$$\log(a(x, \mathbf{y}) - 0.5) = 1 + y_1 (\sqrt{\pi} R/2)^{1/2} + \sum_{n=2}^N \zeta_n \varphi_n(x) y_n, \quad (5.4a)$$

where

$$\zeta_n := (\sqrt{\pi} R)^{1/2} \exp\left(\frac{-(\lfloor n/2 \rfloor \pi R)^2}{8}\right), \quad n > 1 \quad (5.4b)$$

and

$$\varphi_n(x) := \begin{cases} \sin\left(\frac{\lfloor n/2 \rfloor \pi x_1}{R_p}\right), & n \text{ even} \\ \cos\left(\frac{\lfloor n/2 \rfloor \pi x_1}{R_p}\right), & n \text{ odd.} \end{cases} \quad (5.4c)$$

The random variables $\{y_n\}_{n=1}^N$ are i.i.d. and are each uniformly distributed in $[-\sqrt{3}, \sqrt{3}]$, with zero mean and unit variance, i.e., $\mathbb{E}[y_n] = 0$, and $\mathbb{E}[y_n y_m] = \delta_{nm}$, for $n, m \in \mathbb{N}_+$. The finite dimensional stochastic diffusion a represents the N -term truncation of an expansion of a random field with stationary covariance function, given by

$$\text{Cov}[\log(a - 0.5)](x_1, x'_1) = \exp\left(-\frac{(x_1 - x'_1)^2}{R_c^2}\right), \quad (5.5)$$

where $x_1, x'_1 \in [0, 1]$, and R_c is the physical correlation length for the random field a . The parameter R_p in (5.4c) is given by $R_p = \max\{1, 2R_c\}$ and R is given by $R = R_c/R_p$. Then ζ_n and $\varphi_n(x)$ are the eigenvalues and eigenfunctions associated with (5.5). Here we will consider two correlation lengths, namely $R_c = 1/2$, and $R_c = 1/64$, where Figure 5.2 shows the corresponding decay of eigenvalues. For the spatial discretization, we use a finite element approximation on a regular triangular mesh with linear finite elements and 4225 degrees of freedom. The CG method is used for the linear solver with diagonal preconditioners and a tolerance of 10^{-14} .

First, for $R_c = 1/64$, the error and total iteration count of both the standard case, using zero initial vectors, and accelerated SC construction, computed using several dimensions N , are summarized in Table 5.2. The error is measured using the expectation of the approximate solutions, $\|\mathbb{E}[u_{h, L_{\max}}] - \mathbb{E}[u_{h, L^*}]\|_{L^2(D)}$, for $L_{\max} = 1, \dots, 7$, where the “exact” solution $\mathbb{E}[u_{h, L^*}]$ is computed using $L^* = 8$. We compare these errors against the cumulative total number of iterations, K_{zero} and K_{acc} , needed to construct $\mathbb{E}[u_{h, L_{\max}}]$.

An alternative approach to accelerating SC methods is found in [21]. For a particular SC level L_{\max} , this method orders the collocation points lexicographically, with each dimension ordered according to the decay of the eigenvalues in (5.4a). We also implemented a similar method without the sequential ordering; for a given level L , at each new collocation point in $\Delta\mathcal{H}_L$ the solution at the nearest collocation point from lower levels is given as an initial guess to accelerate the CG solver. We refer to this method as the “nearest neighbor” approach. Figure 5.3 shows the average number of iterations needed to solve the linear system (3.7), where the average is taken over the new points at level L , i.e., $\Delta\mathcal{H}_L$, for $L = 1, \dots, 7$. We compare our interpolated acceleration algorithm, the

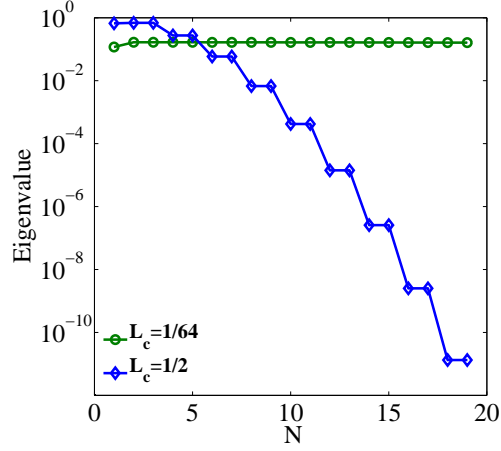


Fig. 5.2: First 19 eigenvalues for (5.5) for correlation length $R_c = 1/64, 1/2$.

	Error	SC Pts	K_{zero}	K_{acc}	Savings in K
N=3	3.83e-8	25	6,780	5,991	11.6%
	9.57e-10	69	18,893	14,628	22.6%
	9.86e-12	177	48,691	27,765	43.0%
N=5	5.28e-07	61	17058	15095	11.6%
	1.03e-08	241	67,955	53,992	20.6%
	1.44e-10	801	226,597	150,241	33.7%
N=7	2.43e-08	589	168,237	136,072	19.1%
	6.63e-10	2,465	706,049	500,718	29.1%
	1.94e-11	9,017	2,585,970	1,496,391	42.1%
N=9	1.68e-07	1,177	338,428	277,583	18.0%
	7.83e-09	6,001	1,729,337	1,273,895	26.3%
	8.86e-11	26,017	7,505,343	4,719,820	37.1%
N=11	2.59e-07	2,069	596,368	495,705	16.9%
	2.43e-08	12,497	3,608,185	2,736,615	24.2%
	1.95e-09	63,097	18,231,420	12,139,658	33.4%

Table 5.2: Iteration counts and savings of the accelerated SC method for solving (5.3)–(5.4) with correlation length $R_c = 1/64$, and stochastic dimensions $N = 5, 7, 9$, and 11.

nearest neighbor approach, and standard SC method without acceleration, for $N = 3$ and $N = 11$, using $R_c = 1/64$. The interpolated initial vector provided by the acceleration algorithm yields a reduction in the average number of iterations at each level, which increases with L . Figure 5.3 also shows the effect of using the nearest neighbor solution as the initial vector, which provides some improvement over the standard case using zero initial vectors, but the savings do not match those of our acceleration scheme. Note that since the number of new collocation points grows exponentially

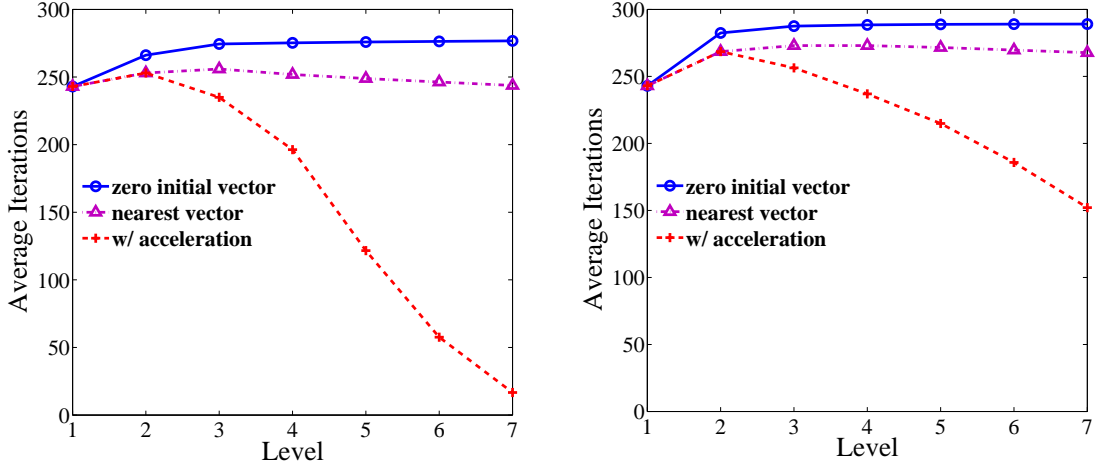


Fig. 5.3: Comparison of the average CG iterations per level for solving problem (5.3)–(5.4) with dimensions $N = 3$ (left) and $N = 11$ (right), and correlation length $R_c = 1/64$.

with each level (cf (4.3)), there is an increase in *total* iteration savings over successive levels in both the nearest neighbor and accelerated case.

The left plot of Figure 5.4 shows the total iteration savings achieved by the acceleration algorithm with different maximum collocation levels $L_{\max} = 1, \dots, 6$. The savings are measured as the percentage reduction in the cumulative iteration count up to level L_{\max} , relative to standard case using zero initial vectors, i.e., $(K_{\text{zero}} - K_{\text{zero}})/K_{\text{zero}}$. Here we also see the effect of stochastic dimension on the convergence of SC methods: as N increases, our algorithm provides less accurate initial guesses for a given maximum SC level L_{\max} . This can also be seen by comparing the left and right plots of Figure 5.3, which show how the average number of iterations at a given SC level L changes from $N = 3$ to $N = 11$. On the other hand, the right plot of Figure 5.4 shows the same total iteration savings now plotted versus error. As above, the error is measured as $\|\mathbb{E}[u_{h,L_{\max}}] - \mathbb{E}[u_{h,L^*}]\|_{L^2(D)}$, with $L^* = 7$. These results are in agreement with the theoretical asymptotic estimates from Theorem 4.8, which predict an increased savings vs error for larger dimensions.

Next we examine the effect of the correlation length, R_c , on our acceleration algorithm. Larger correlation lengths result in faster decay of eigenvalues of the covariance function (5.5) (see Figure 5.2), and implies that $u(\mathbf{y})$ depends on certain components of the vector \mathbf{y} more than others, which reduces the effectiveness of isotropic methods. Figure 5.5 plots the convergence of the error in $\mathbb{E}[u_{h,L}]$ versus the total number of CG iterations for $N = 3$ and $N = 11$, and for both $R_c = 1/2$ and $R_c = 1/64$. The larger correlation length, $R_c = 1/2$, results in slower convergence of the SC interpolant than for $R_c = 1/64$, but note that the accelerated method reduces the total iteration count in both cases.

On the other hand, we can employ *anisotropic* methods to increase the efficiency of SC in the case of larger correlation lengths [30]. Anisotropic SC methods will place more points in directions corresponding to large eigenvalues of (5.5), and the importance of each dimension is encoded in a weight vector (see (4.3)). Figure 5.6 plots the average number of iterations for problem (5.3)–(5.4) with a relatively large correlation length $R_c = 1/2$, and $N = 11$. Here we employ the weights given by an *a posteriori* selection described in [30], i.e., the weight vector $\boldsymbol{\alpha} \in \mathbb{R}^N$, with $\alpha_1 = 0.85, \alpha_2 =$

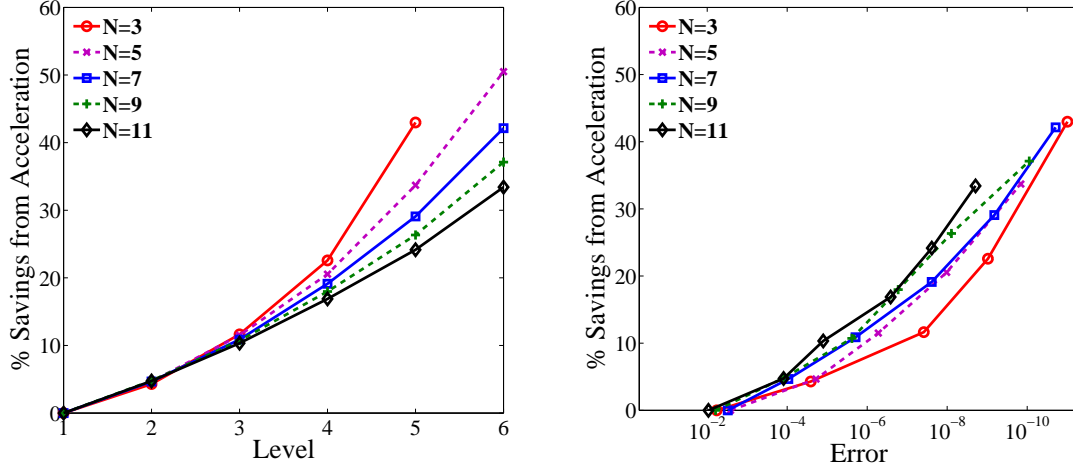


Fig. 5.4: Percentage cumulative reduction in CG iterations vs level (left) and error (right) for solving (5.3)–(5.4) using our accelerated approach, with $N = 5, 7, 9$, and 11 and for correlation length $R_c = 1/64$.

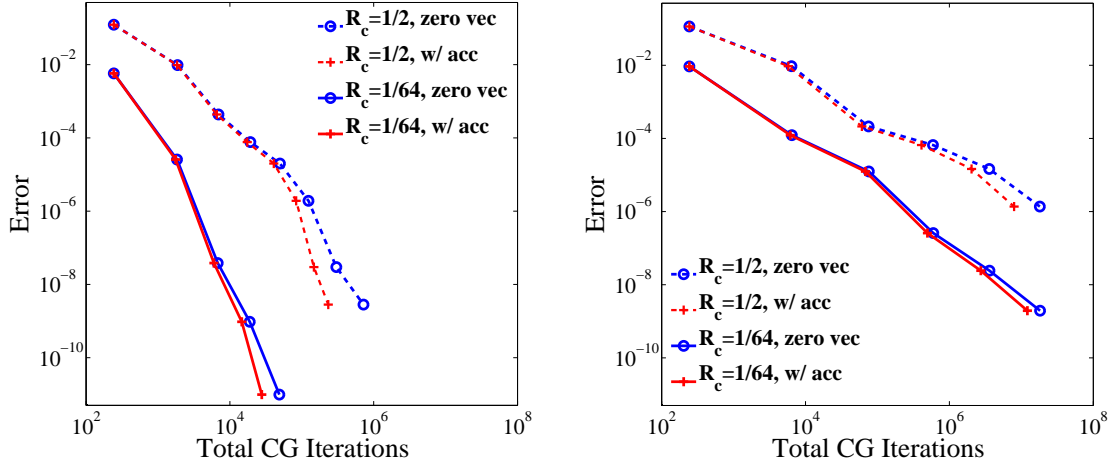


Fig. 5.5: The convergence of the SC approximation for solving (5.3)–(5.4), using CG, with and without acceleration, for correlation lengths $R_c = 1/64, 1/2$, and dimensions $N = 3$ (left), and $N = 11$ (right).

$\alpha_3 = 0.8, \alpha_4 = \alpha_5 = 1.0, \alpha_6 = \alpha_7 = 1.6, \alpha_8 = \alpha_9 = 2.6, \alpha_{10} = \alpha_{11} = 3.7$. The acceleration method decreases the average number of iterations needed to solve the linear system, but the effect is not as pronounced as in the case of an isotropic SC method. This occurs because the isotropic method places far too many points in relatively unimportant directions, thus the dependence of $u(\mathbf{y})$ on a certain component y_n of \mathbf{y} may be well approximated at very low levels. Anisotropic methods

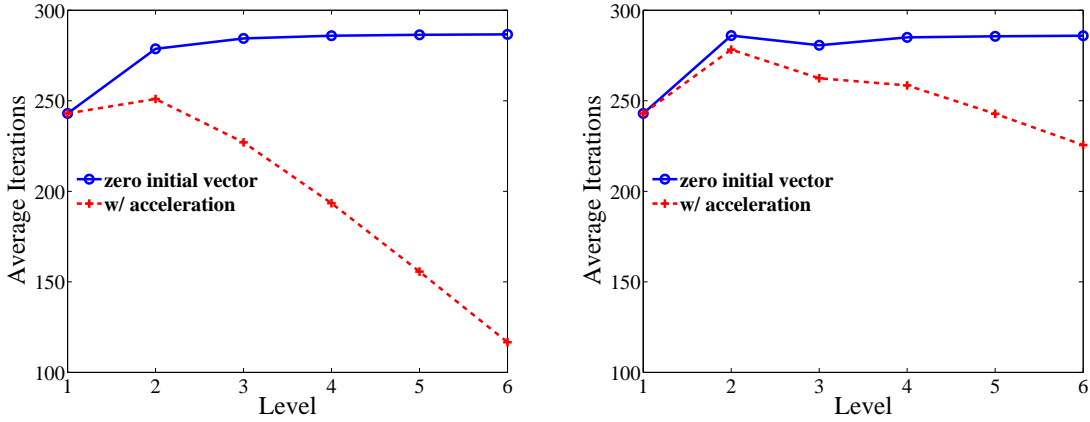


Fig. 5.6: Average CG iterations per level for solving problem (5.3)–(5.4) for $N = 11$ and with correlation length $R_c = 1/2$, using an isotropic SC (left) and anisotropic SC (right). The inefficiencies from using an isotropic grid are partially offset by increased gains from acceleration.

exhibit better convergence with respect to $M_{L_{\max}}$ (and lower interpolation costs) versus isotropic methods, yet we see here that the acceleration algorithm helps to somewhat offset the inefficiency of isotropic methods for anisotropic problems.

In the preceding results we have used a simple diagonal preconditioner strategy. As described in Remark 3.3, we can also construct efficient preconditioners with our acceleration scheme. Table 5.3 shows the effectiveness of the preconditioning strategy for solving equations (5.3)–(5.4), with $N = 7$ and $R_c = 1/64$, where we compare the average number of iterations needed to solve (3.7) at each new point $\mathbf{y}_{L,j} \in \Delta\mathcal{H}_L$ at a given level L . Here we compute an incomplete Cholesky preconditioner for each linear system on the levels $L = 1, \dots, L_{\text{PC}}$, for $L_{\text{PC}} = 1, 2$, and 3, and use these to provide an “accelerated” preconditioner (3.11) for the systems on the remaining levels $L_{\text{PC}} + 1, \dots, L_{\max}$. We compare this against the cases where a simple diagonal preconditioner and an incomplete Cholesky preconditioner are used for each system. The three-level accelerated preconditioner reduces the average number of iterations to within a decimal point of the incomplete Cholesky preconditioner, and the cost of computing the low-level preconditioners and interpolating is relatively cheap in comparison.

Example 5.3. The preceding experiments demonstrate the benefits of using acceleration to improve the convergence of individual iterative linear solvers. In the case of a nonlinear PDE, the possibilities for savings can be even greater than the linear cases above, since convergence of a nonlinear solver may be slow or even unattainable from a poor initial vector. In this example, we consider the problem

$$\begin{cases} -\nabla \cdot (a(x, \mathbf{y}) \nabla u(x, \mathbf{y})) + F[u](x, \mathbf{y}) = x & \text{in } D \times \Gamma, \\ u(0, \mathbf{y}) = 0 & \text{in } \Gamma, \\ u'(1, \mathbf{y}) = 1 & \text{in } \Gamma, \end{cases}$$

where a is given by (5.2), $D = [0, 1]$, $\Gamma_n = [-1, 1]$, $n = 1, \dots, 4$, and $F[u]$ is some nonlinear function of u . In what follows, we consider the nonlinear functions $F[u] = u^5$, and $F[u] = uu'$.

CG iterations for standard SC						
Level	No PC	Diag PC	Inc. Chol.	$L_{PC} = 1$	$L_{PC} = 2$	$L_{PC} = 3$
1	243	243	55	55	–	–
2	311.8	278.4	54.7	60.7	54.7	–
3	332.3	284.9	54.6	63.5	54.9	54.6
4	341.0	286.1	54.6	65.2	55.3	54.6
5	345.8	286.7	54.6	66.2	55.5	54.6
6	348.4	286.9	54.6	66.7	55.6	54.6

CG iterations for accelerated SC						
Level	No PC	Diag PC	Inc. Chol.	$L_{PC} = 1$	$L_{PC} = 2$	$L_{PC} = 3$
1	243	243	55	55	–	–
2	299.3	264.6	52.9	58.4	52.9	–
3	295.8	251.3	49.1	57.1	49.4	49.1
4	270.8	225.8	43.7	52.3	44.2	43.7
5	237.0	194.3	37.3	45.8	38.0	37.3
6	186.1	151.9	28.9	36.0	29.5	28.9

Table 5.3: Average iteration counts for the standard SC method (top), and the accelerated SC method (bottom) using six preconditioner schemes to solve (5.3)–(5.4) with $N = 7$, and $R_c = 1/64$. From left to right: no preconditioner, diagonal preconditioners, incomplete Cholesky preconditioners, and accelerated preconditioners (3.11) built using incomplete Cholesky preconditioners with $L_{PC} = 1, 2, 3$.

Nonlinear problems are typically solved with the use of iterative methods such as Picard iterations or Newton’s method. We implement a combination of these methods that begins with Picard iterations, then utilizes Newton’s method once the relative errors are small. For spatial discretization, we use piecewise linear finite elements on $[0, 1]$ with a mesh size of $h = 1/500$, and solved the resulting systems at each iteration using exact methods. The stopping criterion for the solver is a relative tolerance of 10^{-8} in the l^2 norm.

Results for these experiments are given in Figure 5.7. For each SC level, $L = 1, \dots, 8$, we plot the average number of nonlinear iterations, where the average is taken over the set of points which are new to level L , namely $\Delta\mathcal{H}_L$. Finally, we show the total computational time in Table 5.4, for different maximum levels of stochastic approximation, measured on a workstation with 1.7GHz dual core processors and 8 GB of RAM. We note that in Table 5.4, the size of the finite element system is fixed. Thus, as we move to higher levels of collocation, the stochastic approximation becomes relatively more expensive to compute compared to the solving the finite element systems. This is why the savings begin to decrease after level 5, even though Figure 5.7 shows dramatic savings in iterations for higher levels. Furthermore, the reason for the negative savings for a level $L = 2$ stochastic approximation is that the interpolant is not yet accurate enough to overcome the additional cost of the acceleration.

6. Conclusion. In this work, we proposed and analyzed an acceleration method for construction of sparse interpolation-based approximate solutions to PDEs with random input parameters. The acceleration method exploits the sequence of increasingly accurate approximate solutions to provide increasingly good initial guesses for the underlying iterative solvers that are used at each sample point. We have developed this method using a global Lagrange polynomial basis but the

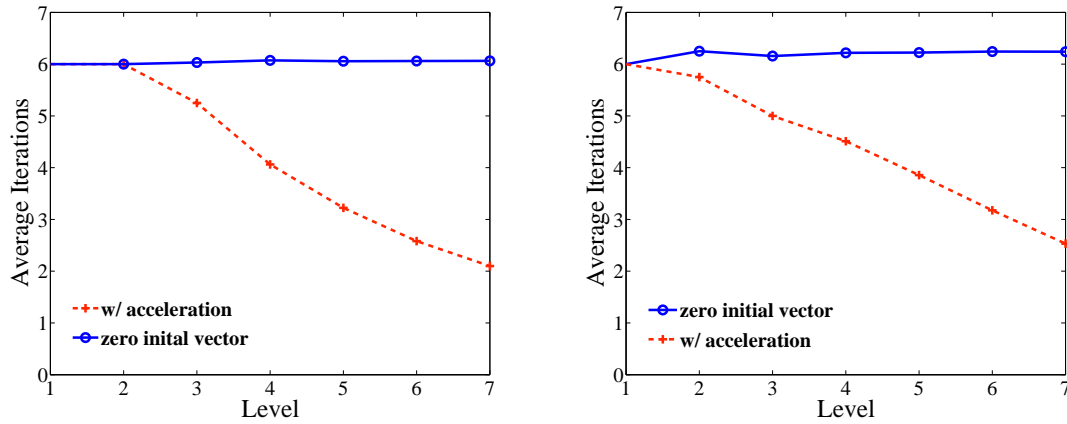


Fig. 5.7: Average number of Newton iterations per level for solving problem (5.6) with $F[u] = uu'$ (left) and $F[u] = u^5$ (right).

SC Level	2	3	4	5	6
$F[u] = u^5$, acc	.03018	.113832	.2746	.7039	2.33314
$F[u] = u^5$, zero	.025976	.119256	.339678	.949184	2.61958
% Savings	-16.2	4.5	19.2	25.8	10.9
$F[u] = uu'$, acc	.027754	.089082	.22706	.629451	2.05741
$F[u] = uu'$, zero	.026527	.090435	.273355	.895027	2.4008
% Savings	-4.6	1.5	16.9	29.7	14.3

Table 5.4: Computational time in seconds for computing solution to problem 5.6.

method can easily be extended to other non-intrusive methods.

While our method takes advantage of the natural structure provided by hierarchical SC methods, we do not take advantage of any hierarchy in the spatial approximation. As explained in Remark 3.1, our method may be used in combination with the multilevel method to accelerate the construction of stochastic operators, and reuse information from level to level. The combination of the acceleration scheme with multilevel methods will be the subject of future work.

We rigorously studied error estimates in the special the case of linear elliptic PDEs with random inputs, providing complexity estimates for the proposed method. Several numerical examples confirm the expected performance. While the analysis of §4.1 applies to linear stochastic PDEs, the acceleration method may be even more well suited to nonlinear problems, as convergence rates may be improved, based on the choice of a good initial guess for nonlinear iterative solvers. A final numerical example demonstrates the advantage of our approach to nonlinear problems. A more rigorous study of acceleration for nonlinear solvers and extension to time dependent problems may provide interesting opportunities in the future.

References.

- [1] I. BABUŠKA, F. NOBILE, AND R. TEMPONE, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007),

- pp. 1005–1034.
- [2] J. BÄCK, F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: A numerical comparison*, in Spectral and High Order Methods for Partial Differential Equations, J. S. Hesthaven and E. M. Rønquist, eds., vol. 76 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
 - [3] R. E. BANK AND L. R. SCOTT, *On the conditioning of finite element equations with highly refined meshes*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 1383–1394.
 - [4] A. BARTH, A. LANG, AND C. SCHWAB, *Multilevel Monte Carlo method for parabolic stochastic partial differential equations*, BIT Numerical Mathematics, 53 (2013), pp. 3–27.
 - [5] A. BARTH, C. SCHWAB, AND N. ZOLLINGER, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numer Math, 119 (2011), pp. 123–161.
 - [6] J. BECK, F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients*, Computers and Mathematics with Applications, 67 (2014), pp. 732–751.
 - [7] J. BURKARDT, M. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *A hyper-spherical sparse grid approach for high-dimensional discontinuity detection*, SIAM Journal on Numerical Analysis, (2014). To appear.
 - [8] T. F. CHAN AND M. K. NG, *Galerkin projection methods for solving multiple linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 836–850.
 - [9] A. CHKIFA, A. COHEN, AND C. SCHWAB, *Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs*, Journal de Mathématiques Pures et Appliquées, (2014).
 - [10] C. W. CLENSHAW AND A. R. CURTIS, *A method for numerical integration on an automatic computer*, Numerische Mathematik, 2 (1960), pp. 197–205.
 - [11] K. A. CLIFFE, M. B. GILES, R. SCHEICHL, AND A. L. TECKENTRUP, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), pp. 3–15.
 - [12] A. COHEN, R. DEVORE, AND C. SCHWAB, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs*, Analysis and Applications, 9 (2011), pp. 11–47.
 - [13] S. DE MARCHI, *On Leja sequences: some results and applications*, Applied mathematics and computation, 152 (2004), pp. 621–647.
 - [14] R. DEVORE, H. TRAN, C. WEBSTER, AND G. ZHANG, *Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients*, (2015). To appear.
 - [15] V. K. DZJADYK AND V. V. IVANOV, *On asymptotics and estimates for the uniform norms of the Lagrange interpolation polynomials corresponding to the Chebyshev nodal points*, Analysis Mathematica, 9 (1983), pp. 85–97.
 - [16] O. G. ERNST AND E. ULLMANN, *Stochastic Galerkin matrices*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1848–1872.
 - [17] G. FISHMAN, *Monte Carlo: concepts, algorithms, and applications*, Springer, 1996.
 - [18] R. G. GHANEM AND R. M. KRUGER, *Numerical solution of spectral stochastic finite element systems*, Computer Methods in Applied Mechanics and Engineering, 129 (1996), pp. 289–303.
 - [19] R. G. GHANEM AND P. D. SPANOS, *Stochastic finite elements: a spectral approach*, Courier Dover Publications, 2003.
 - [20] M. B. GILES, *Multilevel Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.
 - [21] A. D. GORDON AND C. E. POWELL, *On solving stochastic collocation systems with algebraic*

- multigrid*, IMA Journal of Numerical Analysis, 32 (2012), pp. 1051–1070.
- [22] M. D. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *An adaptive wavelet stochastic collocation method for irregular solutions of partial differential equations with random input data*, in Sparse Grids and Applications-Munich 2012, Springer, 2014, pp. 137–170.
 - [23] ———, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numerica, 23 (2014), pp. 521–650.
 - [24] J. HELTON AND F. DAVIS, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems*, Reliability Engineering and System Safety, 81 (2003), pp. 23–69.
 - [25] C. JIN, X.-C. CAI, AND C. LI, *Parallel domain decomposition methods for stochastic elliptic equations*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2096–2114.
 - [26] M. LOÈVE, *Probability theory. I*, vol. 45 of Graduate Texts in Mathematics, Springer-Verlag, 1977.
 - [27] G. MIGLIORATI, F. NOBILE, E. SCHWERIN, AND R. TEMPONE, *Approximation of quantities of interest in stochastic PDEs by the random discrete l^2 projection on polynomial spaces*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1440–A1460.
 - [28] ———, *Analysis of discrete l^2 projection on polynomial spaces with random evaluations*, Foundations of Computational Mathematics, 14 (2014), pp. 419–456.
 - [29] H. NIEDERREITER, *Random number generation and quasi-Monte Carlo methods*, vol. 63 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
 - [30] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2411–2442.
 - [31] ———, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2309–2345.
 - [32] B. ØKSENDAL, *Stochastic differential equations*, Springer, 2003.
 - [33] M. L. PARKS, E. DE STURLER, G. MACKEY, D. D. JOHNSON, AND S. MAITI, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1651–1674.
 - [34] M. F. PELLISSETTI AND R. G. GHANEM, *Iterative solution of systems of linear equations arising in the context of stochastic finite elements*, Advances in Engineering Software, 31 (2000), pp. 607–616.
 - [35] C. E. POWELL AND H. C. ELMAN, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.
 - [36] A. QUARTERONI, R. SACCO, AND F. SALERI, *Numerical Mathematics*, Texts in applied mathematics, Springer, 2000.
 - [37] Y. SAAD, *Iterative methods for sparse linear systems*, SIAM, 2003.
 - [38] V. SIMONCINI AND D. B. SZYLD, *Recent computational developments in Krylov subspace methods for linear systems*, Numerical Linear Algebra with Applications, 14 (2007), pp. 1–59.
 - [39] M. STOYANOV AND C. G. WEBSTER, *A gradient-based sampling approach for stochastic dimension reduction for partial differential equations with random input data*, International Journal for Uncertainty Quantification, (2014). To appear.
 - [40] A. TECKENTRUP, *Multilevel Monte Carlo methods and uncertainty quantification*, PhD thesis, University of Bath, 2013.
 - [41] A. L. TECKENTRUP, P. JANTSCH, C. G. WEBSTER, AND M. GUNZBURGER, *A multilevel stochastic collocation method for partial differential equations with random input data*, Tech. Rep. ORNL/TM-2014/621, Oak Ridge National Laboratory, 2014.
 - [42] L. N. TREFETHEN, *Is Gauss quadrature better than Clenshaw-Curtis?*, SIAM review, 50 (2008),

pp. 67–87.

- [43] ———, *Approximation theory and approximation practice*, SIAM, 2013.
- [44] D. XIU AND G. E. KARNIADAKIS, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM Journal on Scientific Computing, 24 (2002), pp. 619–644.